Doctoral Dissertation Academic Year 2013

Analysis of Peer-to-Peer Operation in Content Delivery.

Keio University
Graduate School of Media and Governance

Mohamad Dikshie Fauzie

Summary

Live video streaming has long been projected as the killer application for Internet. In recent years with the deployment of increased bandwidth in the last mile, this promise finally turned into reality. There are competing technologies to deliver live video streaming: CDN (content delivery network) and P2P (peer-to-peer). CDNs provides end-users with the appearance of traditional client server approaches but enable content providers to handle much larger request volumes. At the same time, ISPs can also benefit from deploying CDN servers in their networks as it reduces the total amount of upstream and transit traffic. CDN provide excellent quality to end-users when the workload is within provisioning limits. P2P systems solve the scalability issue by leveraging the resources of the participating peers, while keeping the server requirement low. However, decentralized uncoordinated of P2P operation comes with undesirable side effects: unfairness in the face of heterogeneous peer resources, network unfriendliness, etc. On the other hand, the growth of video traffic is also contribute to increases of power consumption and it's need to be considered.

In this research, Bittorrent as one of the most popular and successful P2P applications in the current Internet is taken as example the study of uncoordinated P2P operation. First problem to be addressed in this research is how to reveal the topology of real Bittorrent swarms, how dynamic the topology is, and how it affects overall behavior. We study of Bittorrent networks, where real-world Bittorrent swarms were measured using a rigorous and simple method in order to understand the Bittorrent network topology. We propose the Bittorrent Peer Exchange (PEX) messages to infer the topology of Bittorrent swarms listed on a Bittorrent tracker claiming to be the largest Bittorrent network on the Internet, instead of building small Bittorrent networks on testbeds such as PlanetLab and OneLab as other researchers have done. We also performed simulations using the same approach to show the validity of the inferred topology resulted from the PEX messages by comparing it with the topology of the simulated network. Our result, verified using the Kolmogorov-Smirnov goodness of fit test and the likelihood ratio test and confirmed via simulation, show that a power-law with exponential cutoff is a more plausible model than a pure power-law distribution. We also found that the average clustering coefficient is very low, implies the the Bittorrent swarms are close to random networks. Bittorrent swarms are far more dynamic than has been recognized previously, potentially impacting attempts to optimize the performance of the system as well as the accuracy of simulations and analyses.

In the current content delivery architecture, many CDN companies and ISPs adopt hybrid CDN-P2P because the advantage of P2P. In P2P side, peers are organized in a tree based overlay on a per substream basis for live streaming. This ensure that all peers contribute some upload bandwidth. Each CDN server keeps track of clients currently assigned to it to avoid undesirable side effects of P2P. Each client learns about other peers assigned to its designed CDN server. Since in hybrid CDN-P2P architecture some of workload or data delivery are done by peers, therefore CDN server foreseeing the potential power consumption reduction. Second problem to addressed in this research is what's the trade-off of hybrid CDN-P2P architecture compare to CDN. We solve this problem by proposing simple model of power consumption of CDN server and router including the cost of cooling that needed generated from power consumption of CDN server and router. Furthermore, this power reduction can be used for capacity planning of data center.

Finally, proposed methodology can contribute largely to further characterizing P2P networks and promotion of relaxing capacity planning data center in term of energy consumption by hybrid CDN-P2P.

Acknowledgements

This page goes to ACK

Contents

List of Figures

Chapter 1

Introduction

1.1 Backround

Internet-based multimedia content delivery enables users to watch desired content from any location at an any point of time. With the increasing capacities of end-user devices and faster Internet connections, the popularity of such services is growing steadily. Cisco VNI predicts that video streaming will significantly outweigh other types of consumer Internet traffic, such as file sharing, Web, Voice over IP (VoIP), and online gaming [7]. Contrary to file transfers, video streaming enables users to watch the video while downloading it, which imposes strict requirements on the delivery infrastructure. The users expect a performance similar to the traditional television with short startup delays and without performance degradations or playback stalling during watching. This is exacerbated by the growing requirements on video quality, such as higher resolutions and additional features (high-definition and 3D videos). The higher quality typically results in increased video bitrates that require higher download bandwidth. Contrary to file transfers, video streaming enables users to watch the video while downloading it, which imposes strict requirements on the delivery infrastructure. The users expect a performance similar to the traditional television with short startup delays and without performance degradations or play- back stalling during watching. This is exacerbated by the growing requirements on video quality, such as higher resolutions and additional features (high-definition and 3D videos). The higher quality typically results in increased video bitrates that require higher download bandwidth.

Today users are increasingly able to consume videos directly from their TV screens using Internet-enabled Set-top Boxes (STBs) such as digital video recorders, game con- soles, or other entertainment devices. Questions arise as which delivery architecture is able to provide this vast amount of video content to end-user devices and which mechanisms are required to make this architecture scalable and cost-efficient. Common solutions are centralized and decentralized delivery architectures employing various mechanisms to deliver video streams to end-users. These delivery architectures build overlay networks on top of the underlying Internet infrastructure. The simplest architecture for video streaming is based on the centralized client-server model. Here (one or many) video servers send a separate video stream to each client, which results in high bandwidth costs for popular content and potential scalability issues for large numbers of concurrent users. The peer-to-peer (p2p) paradigm offers a promising alternative to pure server-based video distribution networks. Here, the users, called peers1, not only consume but also provide services to other peers. The application of the p2p paradigm to video streaming uses peers resources, such as local storage, computational power, and band- width, to reduce the load and costs of content servers. In the extreme case of pure p2p streaming, there are no dedicated servers anymore and all services are provided by regular peers. If we consider a commercial streaming system, a pure p2p solution turns out to be insufficient because it lacks important properties such as service guarantees for users, security, and control by the content provider. In order to overcome these limitations, a peer-assisted architecture can combine content servers and peers intelligently.

In order to understand the relationships between entities in P2P-CDN ecosystem and to identify the possible tensions, we must understand their roles in the architecture:

- Overlay providers contribute the initial content and host servers for content injection and indexing. In a pure commercial scenario the overlay provider also acts as a content provider, while in a scenario with user-generated content the content is contributed by users that upload it to content servers. Typically, an overlay provider receives certain payments for the hosted content, either directly from users (usage-dependent or subscription-based) or indirectly via advertisements.

- Users consume the streamed videos but also provide their resources to the system, such as upload bandwidth, local storage space, and online time. The users typically pay flat-rate fees for the Internet access, which explains why they allow an overlay provider to use their upload bandwidth.

- Network operators provide infrastructure for Internet access and receive flat-rate payments from the users for this service. Typical delivery overlays span several network domains controlled by different network operators. Therefore, network operators must manage both the internal and external traffic flows to avoid congestion and excessive payments for traffic transit.

A peer-assisted solution shifts the main load of content delivery from the overlay providers servers to users. However, the actual delivery costs are shifted from the overlay providers to the network operators. The reason is the widespread acceptance of flat-rate based pricing for the Internet access. These pricing schemes allow network operators to attract users and to sell high-speed Internet connections. But peer-assisted overlays can also lead to bottlenecks and link congestions, since the Internet architecture is built for the client-server traffic pattern where the traffic flows from content servers to the users. In the last years, the management of overlay traffic that crosses the boundaries of network operators domains has gained a lot of attention in the research community. Thereby, various traffic management methods have been proposed to relieve the tension between the overlays and network operators. However, most of them fail to satisfy the demands of both parties.

1.2 Challenges

In this section, we discuss the specific challenges of peer-assisted. Most of these challenges arise from the necessity to achieve quality of service (QoS) comparable to the current client-server systems, while using the limited resources of unreliable peers.

- Limited resources of an individual peer compared to a typical server mean that the resources of many peers must be combined to serve the same streaming request. This applies in particular to the upload bandwidth, which is typically much smaller than the download bandwidth.

- Heterogeneity of peers in terms of their resources and behavior. For example, the upload capacity is a resource that differs between the users and affects the systems performance significantly.

- Lack of service guarantees at peers makes it difficult to ensure a quality streaming experience to the users (comparable to well-dimensioned server-based sys- tems). In a commercial scenario, in contrast to pure p2p-based systems, all users should be able to receive the quality they paid for, which makes the coupling between the peers contribution of resources and received streaming quality undesirable.

- Missing or insufficient incentives for users to contribute their resources are a common issue for p2p-based systems. In a commercial scenario, it can be partially solved by offering rewards or discounts for contributed resources. For example, a peer might get certain credits for each megabyte of data uploaded to other peers. However, this does not solve the issue of users that should remain online in order to provide content availability.

- Energy consumption is becoming an important challenge for content delivery. While various approaches were proposed to increase the energy efficiency of servers and routers in terms of reduced power consumption, the same issue applies for the users devices. One interesting aspect here is whether an idle peer should stay online to serve new requests or leave the system. While the first option would maximize the peers contribution to the system, the second option would save energy that might be wasted if the services of this peer are not required.

- Negative impact on the network infrastructure is another issue in peer-assisted and pure p2p delivery architectures. Most p2p and peer-assisted overlays apply their own application level routing mechanisms that might have undesired effects on the underlying network such as congested links or high costs for the transit traffic.

1.3 Motivation

P2P as in this case BitTorrent as the most popular filesharing applications dominated the Internet traffic and is still growing even thought recent studies suggest that its growth slower than the growth of Internet traffic and its proportion to the Internet traffic is declining. This popularity reflects the robustness and efficiency of the BitTorrent protocol. These characteristics of BitTorrent come from its peer and piece selection strategies to distribute large files efficiently.

Many properties of BitTorrent such as upload, download performance, peer arrival and departure have been studied but only few research have assessed the topological properties of BitTorrent. The BitTorrent system is different from other P22P systems. The BitTorrent protocol does not offer peer traversal and the BitTorrent tracker also does not know about the relationship between peers since peers never sending information to the tracker concerning their connectivity with other peer. While a crawler can be used in other P2P networks such as Gnutella, in BitTorrent we can not easily use a crawler to discover topology, making direct measurement of the topology very difficult and challenging. This BitTorrent swarm topologies reflects peer behavior. The peer relationship behavior is very important as a basis to design controllable P2P system that to be used together with other system for example peer assisted CDN or peer assisted cloud.

In the current modern content delivery, CDN providers tend to combine P2P with CDN in order to help the scaling the services, especially related to traffic or bandwidth saving. It has been done by for example: Akamai and Pando networks. The bigger issue is not traffic or bandwidth that relatively easy to fulfilled by adding network card into router or adding more servers, instead energy consumption by CDN provider itself inside data center. Since the Internet traffic grows, demand for scaling is also grow thus demand for energy is also grow. This growing is constrained by energy supply in data center. The usage of peer assisted CDN can be seen not only for helping scaling the services, but there is potential to reduce the energy consumption furthermore this reduction can relax energy budget inside data center.

1.4 Approach

The key factors in this research are: (1) characterization of peer dynamic in P2P systems and (2) energy consumption trade-off in the using of P2P to assist CDN server.

P2P Swarm Dynamics

The real-world BitTorrent swarms were measured using a rigorous and simple method in order to understand the BitTorrent topology. To our knowledge, our approach is the first to perform such a study on real-world the BitTorrent network topologies. We used the BitTorrent peer exchange (PEX) messages to infer the topology of BitTorrent swarms listed on a BitTorrent tracker claiming the be the largest BitTorrent network on the Internet, instead of building small BitTorrent networks on testbed such as PlanetLab or OneLab as other researchers have done. We also performed simulations using the same approach to show the validity of the inferred topology resulted from the PEX messages by comparing it with the topology of the simulated network.

Energy Consumption of peer assisted CDN

As intermediate step in merging between P2P and CDN, this research introduce energy consumption trade-off in peer-assisted CDN. We analyze the characteristics and the requirements of peer-assisted CDN for live streaming and peer-assisted CDN for online storage. Then we propose energy consumption model for both peer-assisted CDN architectures. To be able to validate the result, we use model from both architectures that currently running on the Internet which are LiveSky [58] and FS2you [12].

1.5 Contribution

This dissertation makes contributions for enabling analysis into integration of P2P services and CDN services.

- For P2P, we propose new and effective methodology to infer BitTorrent swarm topologies. We show that gathering BitTorrent swarm topologies is important as step to understand peer behavior.
- For energy trade-off, we show that both peer-assisted CDN architecture have different in energy characteristics that can be used as considering model in the service integration between P2P and CDN, furthermore this model can use by CDN provider as basis for: capacity planning in data center and incentive planning to customer who will run peer-assisted mode in home gateway.

Chapter 2

Related Work

Bittorrent protocol performance has been explored extensively [16] [26] [37] [47] [27] [59]. Although we know that the topology can have a large impact on performance, to date only a few papers have addressed the issue. Urvoy *et al*. [48] used a discrete event simulator to show that the time to distribute a file in a Bittorrent swarm has a strong relation to the overlay topology. Al-Hamra *et al*. [1], also using a discrete event simulator, showed that Bittorrent creates a robust overlay topology and the overlay topology formed is not random. They also show that peer exchange (PEX) generates a chain-like overlay with a large diameter. Dale *et al*. [9], in an experimental study on PlanetLab, show that in the initial stage of Bittorrent a peer will get a random peer list from the tracker. They found that a network of peers that unchoked each other is scale-free and the node degree follows a power-law distribution with exponent approximately 2. Dale *et al*. [9] also showed that the path length formed in Bittorrent swarms averages four hops and Bittorrent swarms have low average clustering coefficient. However, little work has been done on confirming that such controlled experiments correspond to the system.

We emphasize that compared to Hoßfeld *et al*. [18], our work provides a completely different approach and goal. Hoßfeld *et al*. [18] discuss the AS (Autonomous System) level topology of Bittorrent swarm for optimizing overlay traffic across ASes, while our study focus on microscopic dynamic aspect which is Bittorrent swarms topology itself (peer level or IP address level). The closest work to ours is Kryczka *et al*. [24]. While our method is somewhat similar to theirs, they focus on clustering and locality while our focus is on node degree and clustering. They use PEX to discover peer relationship, unfortunately they do not explain in detail how to process the PEX data set. Because of differing PEX implementations between Bittorrent clients, we need to be careful with it in data processing. In our work, we describe PEX behavior and its limitation on two popular Bittorrent clients: Vuze and uTorrent, and we also explain how to treat PEX data from different Bittorrent clients. We also provide simulation result to confirm that our methods for infering peer relationship with PEX is valid. They observed that Bittorrent swarms have slightly higher clustering coefficient compared to random graphs of the same size and they observe neither Bittorrent swarm fulfills the properties of small world. The slightly difference in clustering come from the difference of PEX data processing. They assume that PEX is the same and complete for all Bittorrent clients, therefore they get slightly different results.

Our results agree with previous research [9] in some areas and disagree in others, perhaps for two reasons. First, power-law claims must be handled carefully. Many steps are required to confirm the power-law behavior, including alternative model checking, and we must be prepared for disappointment since other models may give a better fit. Second, our methodology relies on real work measurement combined with simulation for validation. This real-world measurement will reflect different types of clients connected to our swarm, and each client has a different behavior. We also face difficult-to-characterize network realities such as NAT and firewalls. Our ability to reproduce key aspects of the topology dynamics suggests that these factors have only limited impact on the topology, somewhat to our surprise.

Content Distribution Networks with peer assist have been successfully deployed on the Internet, such as Akamai [21] and LiveSky [58]. The authors of [21] conclude from two real world traces that hybrid CDN-P2P can significantly reduce the cost of content distribution and can scale to cope with the exponential growth of Internet video content. Yin et al. [58] described LiveSkye as commercial operation of a peer-assisted CDN in China. LiveSky solved several challenges in the system design, such as dynamic resource scaling of P2P, low startup latency, ease of P2P integration with the existing CDN infrastructure, and network friendliness and upload fairness in the P2P operation. Measurement from LiveSky showed that LiveSky can save bandwidth around 40% [58]. The author in [20] and [19] proposed mechanisms to minimize CDN server bandwidth to make the content distribution cheap. They designed different peer prefetching policies of video on demand system in surplus mode while ensuring user quality of experience. A similar analysis has been done in [57] for live video streaming system where the authors proposed different limited peer contribution policies to reduce CDN bandwidth requirement and eventually off the distribution process from CDN to P2P system. An ISP friendly rate allocation schemes for a hybrid CDN-P2P video on demand system in [54]. These technique try to minimize CDN server bandwidth while reducing ISP unfriendly traffic and maximizing peer prefetching. Load on CDN server has been shown to be reduced using this approach while reducing cross ISP traffic. Above studies were performed for video on demand or live video streaming. While video is the most popular content, the systems can be also for other type contents. Moreover while content based services are growing, energy consumption of a content distribution system has not been analyzed.

Related to CDN and energy usage, in a seminal work [38], the authors show that if costs are based on electricity usage and if the power prices vary in real-time, global load balancing decision can be made such that users are routed to locations with the cheapest power without significantly impacting user performance or bandwidth cost. In [36], the author proposed utilizing batteries for CDN for reducing total supplied power and total power costs. The authors in [36] also proposed battery provisioning algorithms based on workload of CDN server. The result shows that batteries can provide up to 14% power savings.

The idea that utilize ISP controlled home gateway to provide computing and storage services and adopts managed peer-to-peer model is presented in [51]. Valancius et al. [51] show that distributing computing platform in NaDa (Nano Data Center) save at least 20-30% energy compare to traditional data centers. The saving in NaDa comes from underutilizing home gateway, avoidance of cooling costs, and the reduction of network energy consumption as a result of demand and service co-localization.

The comparison between CDN architecture and peer-to-peer architecture are discussed in [4] and [11]. Both authors in [4] and [11] agree that CDN architecture is more energy saving compare to peer-to-peer architecture. Another interesting study of energy efficient in content delivery architectures is presented by Guan et al. [15]. by Guan et al. [15] comparing energy efficient of CDN architecture and CCN architecture. The authors in [15] conclude that CCN is more energy efficient in delivering popular content while the approach with optical bypass is more energy efficient in delivering infrequent accessed content.

To the best of our knowledge, the study of energy in that considering peer-to-peer in CDN architecture is presented in [30]. Mandal et al. [30] mentioned that hybrid CDN-P2P systems can reduce a significant amount energy in the optical core network around 20-40% less energy. The authors only considered energy consumption of hardware especially optical devices and does not include overhead inside data center, CDN server energy comsumption, and consumed power by peers. Our work is quite different, we take number of peers with static content and add overhead of data center which is power of cooling cause by temperature of hardware for different purpose which are live streaming and video on demand.

Chapter 3

P2P measurement

3.1 Introduction

BitTorrent as the most P2P filesharing application is responsible for a major portion of the Internet traffic share and is daily used by hundred of millions of users. This has attracted the interest of the research community that has thoroughly evaluated the performance and the demographic aspects of BitTorrent. Due to the complexity of the system, the most relevant studies have tried to understand different aspects by performing real measurements of BitTorrent swarms in the wild, this is inferring information from real swarms in real time. Several techniques have been used in order to measure different aspects of BitTorrent so far. In this Chapter we present a survey of different measurement techniques that constitutes a first step in the designing the future measurement techniques and tools for analyzing large scale systems.

3.2 Measuring BitTorrent

In this Section we describe the BitTorrent measurement techniques defined in the literature so far. We classify them into two main categories macroscopic and microscopic depending on the retrieved information.

3.2.1 Macroscopic Technique

The main objective of these techniques is to understand the demographics of the BitTorrent ecosystem: the type of published content, the popularity of the content, the distribution of BitTorrent users per country (or ISP), the relevance of the different portals and trackers, etc. Furthermore, the macroscopic measurements allow to study some performance aspects such as the ratio of seeders/leechers, the session time of the BitTorrent users, the arrival rate of peers, the seedless state (period the torrent is without seeder) duration, etc. We classify the macroscopic techniques into two subcategories: BitTorrent portals crawling and BitTorrent trackers crawling.

3.2.2 Microscopic Technique

The described macroscopic techniques retrieve exclusively the peers IP addresses, thus only metrics associated to the presence/absence of the peer can be studied. Unfortunately, IP address does not suffice to infer relevant performance metrics at the peer level such as peers download and upload rate. For this purpose we need to apply more sophisticated (but less scalable) techniques that we name microscopic techniques. To perform microscopic techniques we need to implement different parts of the BitTorrent peer wire protocol. Any microscopic crawler has to implement the functions to perform the handshaking procedure. This is essential to connect to other peers. The handshaking procedure can be done actively (the crawler initiates it) or passively (the crawler waits until a peer starts the handshaking). Once the crawler is connected to a peer, it exploits different messages of the peer wire protocol in order to measure different parameters.

Chapter 4

Characteristics of BitTorrent Swarms

4.1 Introduction

BitTorrent is one of the most used application in the current Internet and is responsible for an important portion of the upstream and downstream traffic as revealed by recent reports. The significant footprint of BitTorrent in the Internet has motivated researchers and practitioners to dedicate an important amount of effort to understanding and improving BitTorrent. However, despite this effort, we still have little knowledge regarding the connectivity properties exhibited by real BitTorrent swarms at both swarm and peer level. Due to the difficulty in collecting the required information from real swarms, most of the existing works that analyze connectivity properties are based on simulations or experiments in controlled environments. As a result, they are likely to miss some of the effects affecting BitTorrent swarms in the wild.

In this Chapter, we first present a methodology to collect the connectivity information at both the swarm and the peer level for the entire lifespan of a real torrent. Specifically, we discover new torrents just after their birth by using the RSS service of the most important BitTorrent portal, namely the Pirate Bay. Afterwards, we exploit the Peer Exchange (PEX) extension of the BitTorrent protocol to gather the set of neighbors for each peer. PEX is a gossiping technique which main goal is to allow peers to exchange their list of neighbors so that they can learn about other participants in the swarm without contacting the tracker. Note, that PEX has been implemented by most of the existing BitTorrent clients and in particular by the most popular ones such as uTorrent or Vuze. The information collected from PEX (i.e., a peers neighborhood) is the connectivity information at the peer level. Furthermore, by aggregating the neighborhood information collected from every peer in a swarm we are able to build the overlay topology of that swarm (i.e., swarm level connectivity). We retrieve the information from each active peer every 3 minutes and then study the dynamic evolution of both: the overlay topology of the swarm and the composition of each peers neighborhood. We have applied the described methodology to collect the connectivity information of 50 real torrents, including more than 150 peers, since their birth during a period of 10 days.

4.2 Measurement Methodology

The difficulties in inferring topologies in Bittorrent swarms are caused by the Bittorrent mechanism itself. First, although a Bittorrent *peer* may offer some information about its peers, there is no mechanism for peer *discovery*. Second, a peer never sends information about its connections with other peers to the tracker, so we cannot infer overlay topologies by querying or hosting a tracker. Our other options inferring topologies are simulation or deploying Bittorrent nodes in a real network or in a laboratory, e.g., PlanetLab. Deploying hundreds to thousands of nodes in a real network or in the laboratory in a manner that accurately reflects the real world is a very challenging task.

We used PEX to collect information about peer neighbors (see Fig.4.1), and then we describe the network

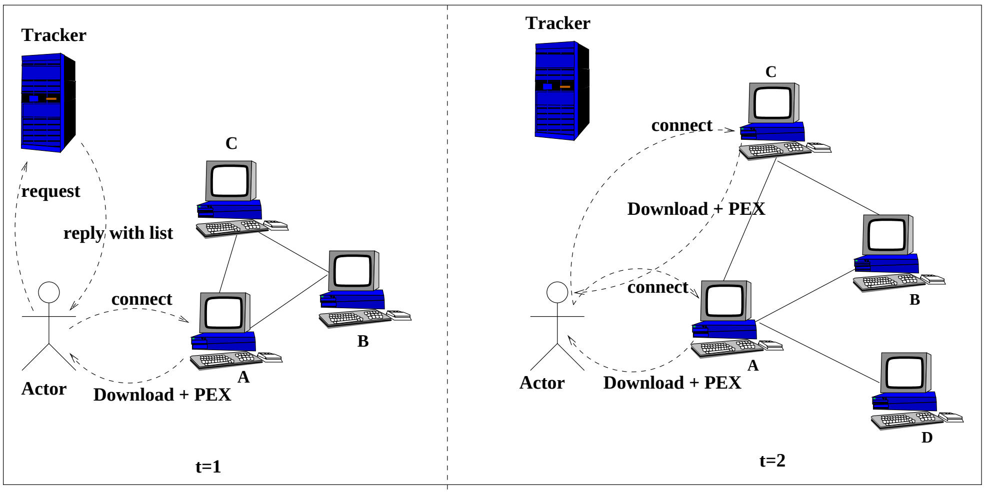

Figure 4.1: Simplified view of our approach. Left: At time t=1, the actor gets a PEX message from peer A and learns that peer A is connected to peer B and C. At t=2, the actor gets PEX messages from peers C and A. The actor learns that now peer A is connected to peer D. Thus the actor knows the properties of peer A at t=1 and t=2.

formed in terms of properties such as node degree and average clustering. Besides collecting data from real Bittorrent networks, we ran simulations similar to these of Al-Hamra *et al.* [2]. In these simulations, we assumed that peer arrivals and departures (churn) follow an exponential distribution as explained by Guo *et al.* [16]. For simplification, we assumed that nodes are not behind a NAT. Since we are only interested in the construction of the overlay topology, we argue that our simulations are thorough enough to explain the overlay properties.

Temporal graphs have recently been proposed to study real dynamic graphs, with the intuition that the behaviour of dynamic networks can be more accurately captured by a sequence of snapshots of the network topology as it changes over time [14] [45]. In highly dynamic networks such as P2P, an instantaneous snapshot may capture only a few nodes and links. In this paper, we study the network dynamics by continuously taking network snapshots over a short time interval Δ, and show them as a time series. A snapshot captures all participating peers and their connections within a particular time interval, from which a graph can be generated. The snapshot duration may have minor effects on analyzing slowly changing networks. However, in a P2P network, the characteristics of the network topology vary greatly with respect to the time scale of the snapshot duration [42]. We consider $\Delta = 3$ minutes to be a reasonable estimate of minimum session length in Bittorrent [40].

4.2.1 Graph Sampling

Due to the large and dynamic nature of Bittorrent networks, it is often very difficult or impossible to capture global properties. Facing this difficulty, sampling is a natural approach. However, collecting unbiased or representative sampling is also sometimes a challenging task.

Suppose that a Bittorrent overlay network is a graph $G(V,E)$ with the peers or nodes as vertices and connections between the peers as edges. If we observe the graph in a time series, i.e., we take samples of the graph, the time-indexed graph is $G_t = G(V_t, E_t)$. From this set of graphs, we can define a measurement window $[t_0, t_0 + \Delta]$ and select peers uniformly at random from the set: $V_{t_0,t_0+\Delta} = \bigcup_{t=t_0}^{t_0+\Delta} V_t$. In that equation, we cannot distinguish properties of the same peer at different times, thus it focuses on sampling peers instead of peer properties. Stutzbach *et al.* [41] showed that equation is only appropriate for exponentially distributed peer session lengths but we know from existing measurement that Bittorrent networks peer session lengths have very high variation [16].

For example: suppose we want to measure number of files shared by peers in Bittorrent swarm. In this

Bittorrent swarm, half of the peers are up all the time and have many files, while other peers remain around for one minute and are immediately replaced by new short-lived peers who have few files. The common method is to observed the system for a long time and sample the peers randomly. This method will incorrectly conclude that most of the peers in the system have very few files. The problem with this method is that it focuses on sampling the number of peers instead of peer properties. Our objective should not be to select a vertex $v_i \in \bigcup_{t=t_0}^{t_0+\Delta} V_t$, but to sample the property of a vertex v_i at a particular instant time t. Therefore, we must distinguish the occurrences of the same peer at different times: samples $v_{i,t}$ and $v_{i,t'}$ gathered at different times $t \neq t'$ are viewed as different, even from the same peer. The key in this method is that we must be able to sample from the same peer more than once at different points in time. Thus we can formalize this into $v_{i,t} \in V_t, t \in [t_0, t_0 + \Delta]$ [41]. With that method, the sampling will not biased because we track the peer's properties overtime.

The number of peers in a swarm that is observed by our client is our population. The sampled peers set is the number of peers that exchange PEX messages with our client. Our sampled peers set through PEX messages exchange can observe about 70% of the peers in a population. This observation is consistent with [56].

4.2.2 Experimental Methodology

We joined the top 50 TV series torrents from the piratebay, which claims to be the biggest torrent tracker on the Internet. Almost all of these torrents were in steady-state phase, which is more dominant than the bootstrapping and decay phases of a torrent's lifetime. We used a modified rasterbar libtorrent [34] client that is connection greedy, where the client tries to connect to all peers it knows without a limit on the number of connections, and the client logs PEX messages received from other clients. PEX messages from old versions of Vuze Bittorrent clients contain all of peers they connected to in the past, hence these clients should be removed from the data. Removal of some peers in data processing is valid in terms of sampling with dynamics, see Sect.(4.2.1). In terms of connectivity, two popular Bittorrent clients (uTorrent and Vuze), by default try to connect to peer candidates randomly without any preference, thus we have random data sets. This implies that our data set is independent of measurement location and the number of measurement locations.

4.2.3 Data Analysis Background

Many realistic networks exhibit the scale-free property [8], though we note that "scale-free" is not a complete description of a network topology [10] [28]. It has been suggested that Bittorrent networks also might have scale-free characteristics [9]. In this paper, we test this hypothesis. Besides testing this hypothesis, we also study the clustering property of Bittorrent swarms.

In a scale-free network, the degree distribution follows a power-law distribution. A power-law distribution is quite a natural model and can be generated from simple generative processes [31], and power-law models appear in many areas of science [8] [31].

A power-law distribution can be described as:

$$Pr[X \geq x] \propto cx^{-\alpha}. \tag{4.1}$$

where x is the quantity of distribution and α is commonly called the scaling parameter. The scaling parameter usually lies in the range $1.8 < \alpha < 3.5$. In discrete form, the above formula can be expressed as:

$$p(x) = Pr(X = x) = Cx^{-\alpha}. \tag{4.2}$$

This distribution diverges on zero, therefore there must be a lower bound of x, called $x_{min} > 0$, that holds for the sample to be fitted by a power-law. If we want to estimate a good power-law scaling parameter then we must also have a good x_{min} estimation.

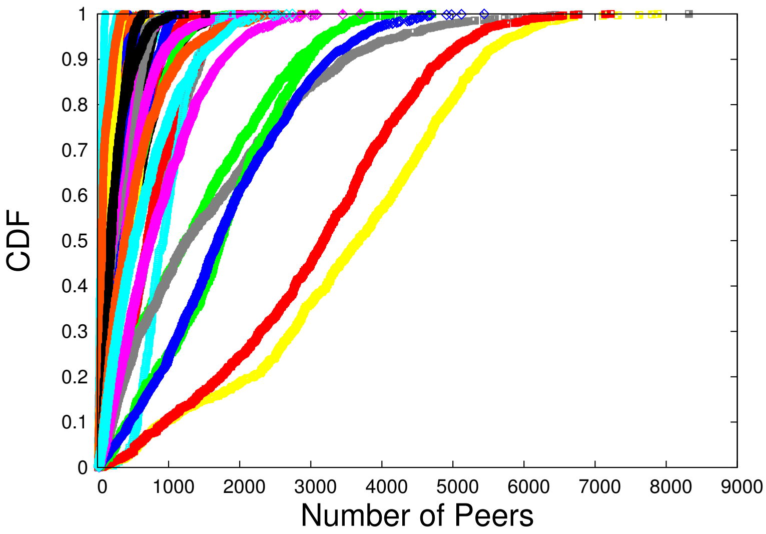

Figure 4.2: CDF plot of number of peers for the 50 swarms during measurement.

Normalizing (4.2) we get

$$p(x) = \frac{x^{-\alpha}}{\zeta(\alpha, x_{min})}. \tag{4.3}$$

The most common way to fit empirical data to a power-law is to take the logarithm of Eq.(4.1) and draw a straight line on a logarithmic plot [31]. We use maximum likelihood to estimate the scaling parameter α of power-law [8]. This approach is accurate to estimate the scaling parameter in the limit of large sample size. For the detailed calculations of both x_{min} and α, see Appendix B in [8].

4.3 Experimental Results

Our time samples for the size of swarms are plotted as the CDF of the number of peers for every swarm during measurement with 104 to 1400 time samples for each torrent, as shown in Fig.4.2. It is clear that the number of peers has high variability due to churn in Bittorrent networks.

4.3.1 Power-law Distribution of Node Degree

The degree of a node in a network is the number of edges connected to that node. If we define p_k as the fraction of nodes in the network that have degree k, then p_k is the probability that a node chosen uniformly at random has degree k. We show node degree data in cumulative distribution function (CDF) plot, which can be expressed as

$$P_k = \sum_{k'=k}^{\infty} p_{k'}. \tag{4.4}$$

We want to know the power-law distribution of the measured Bittorrent networks, and we do not know *a priori* if our data are power-law distributed. To test the applicability of a power-law distribution, we use the goodness-of-fit test as described by Clauset *et al.* [8]. First, we fit data to the power-law model and calculate the Kolmogorov-Smirnov (KS) statistic for this fit. Second, we generate power-law synthetic data sets based on the scaling parameter α estimation and the lower bound of x_{min}. We fit the synthetic data to a power-law model and calculate the KS statistics, then count what fraction of the resulting statistics is larger than the value for the measured data set. This fraction is called the p value. If $p \geq 0.1$ then a power-law model is a good model for the data set, and if $p < 0.1$ then power-law is not a good model.

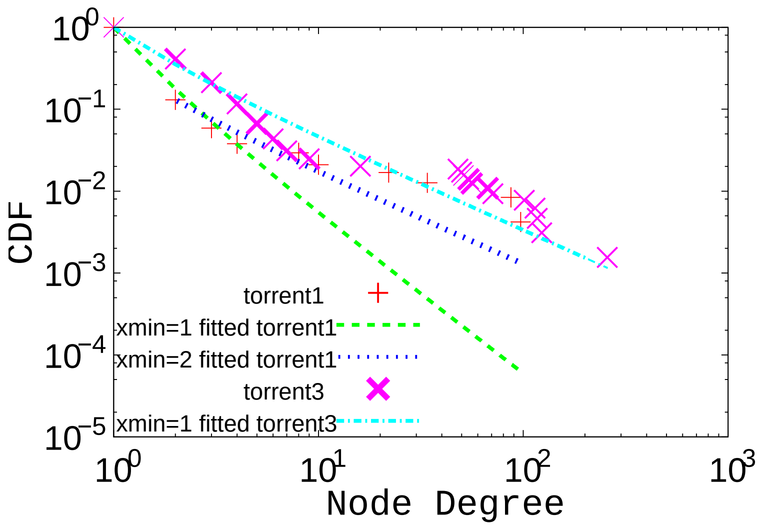

Figure 4.3: Node degree fit for snapshots of two torrents, with three fits shown in log scale.

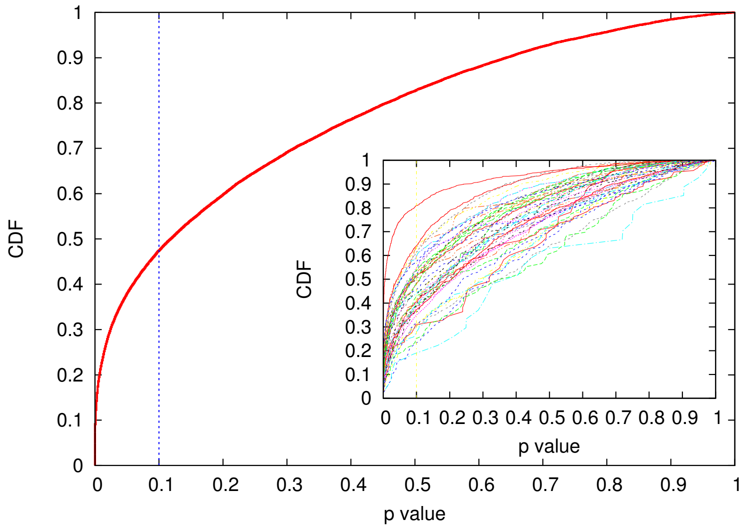

Figure 4.4: CDF plot of p value of K-S statistics.

As mentioned before, a good estimation for x_{min} is important to get a overall good fit. Too small an x_{min} will cause a fit only to the body of the distribution. Too high an x_{min} will cause a fit only to the tail of the distribution. Figure 4.3 illustrates the fit for snapshots of *torrent1* and *torrent3*. For *torrent1*, setting $x_{min} = 2$ leads to $\alpha = 2.11$, while $x_{min} = 1$ gives $\alpha = 2.9$. For *torrent1*, xmin = 1 visually does not give a good fit, while for *torrent3*, setting $x_{min} = 1$ leads to a visually good fit.

Figure 4.4 shows the CDF for p values for all data sets. This figure shows that from the K-S statistics point of view, around 45% of the time a power-law distribution is not a good fit for the data. The inset figure in Fig.4.4 shows the CDF plot p value for each torrent. The dash line on p value = 0.1 is the threshold.

However, these data sets must be interpreted with care. The usage of the maximum likelihood estimators for parameter estimation in power-law is guaranteed to be unbiased only in the asymptotic limit of large sample size, and some of our data sets fall below the rule of thumb for sample size, $n = 50$ [8]. In the goodness-of-fit test, a large p value does not mean the power-law is the correct distribution for data sets, because there may be other distributions that match the data sets and there is always a possibility that even with small value of p the distribution will follow a power-law [8]. We address these concerns next.

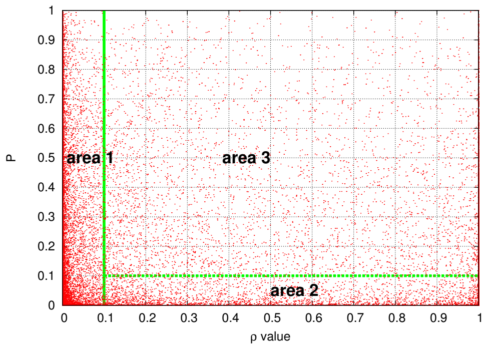

Figure 4.5: Scatter plot of p value vs ρ value.

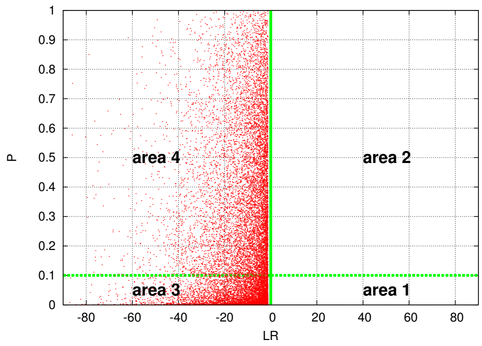

Figure 4.6: Scatter plot of p value vs log-likelihood ratio (LR) for $\rho < 0.1$.

4.3.2 Alternative Distributions

Even if we have estimated the power-law parameter properly and the fit is decent, it does not mean the power-law model is good. It is always possible that non-power-law models are better than the power-law model. There are several methods for direct comparison of two distributions such as *likelihood ratio test* [53], *Bayesian test*, and *Minimum description length*. The likelihood ratio test idea is to compute the likehood of the data sets under two distributions. The one with the higher likehood is the better fit. We use the likelihood ratio test to see whether other distributions can give better parameter estimation.

Nested Case

We now hypothesize that the smaller family of power-law distributions may give a better fit to our data sets. We only consider a power-law model and a power-law with exponential cut-off model as examples to show model selection. Model selection for power-law model and power-law with exponential cut-off is a kind of nested model selection problem. In a nested model selection, there is always the possibility that a bigger family (power-law) can provide as good a fit as the smaller family (power-law with exponential cut-off). In a likelihood ratio test, we must provide the significance value (ρ value). Under the likelihood ratio test, we compare the pure power-law model to power-law with exponential cut-off, and the ρ value here helps us establish which of three possibilities occurs: (i) $\rho > 0.1$ means there is no significant difference between the likelihood of the data under the two hypotheses being compared and thus neither is favored over the other; if we already rejected the pure power-law model, then this does not necessarily tell us that we also can reject the alternative model; (ii) $\rho < 0.1$ and the sign of likelihood ratio (LR) = negative means that there is a significant difference in the likelihoods and that the alternative model is better; if we have already rejected the pure power-law model, then this case simply tells us that the alternative model is better than the bad model we rejected; (iii) if $\rho < 0.1$ and the sign of LR = positive means that there is a significant difference and that the pure power-law model is better than the alternative; if we have already rejected the pure power-law model, then this case tells us the alternative is even worse than the bad model we already rejected.

Figure 4.5 shows a p value vs ρ value scatter plot, divided into three areas. Area 1: $\rho < 0.1$ and $p > 0$. Area 2: $\rho > 0.1$ and $p < 0.1$. Area 3: $\rho > 0.1$ and $p > 0.1$. This figure shows that 52% of the samples lie in area 1, thus an alternative model may be plausible for these samples.

Now we plot p value vs LR as shown in Fig.4.6 for $\rho < 0.1$. We divide the figure into four areas: area 1, area 2, area 3, and area 4 with green lines as borders to see how sparse the points are in each area. Area 1: LR=positive sign and $p < 0.1$. Area 2: LR=positive sign and $p > 0.1$. Area 3: LR=negative sign and $p < 0.1$. Area 4: LR=negative sign and $p > 0.1$. In this figure, 58% of the samples lie in area 3 and 42% lie in area 4, while there are no samples in areas 1 and 2, which means that the alternative model is better. Although in the case $p < 0.1$ we reject power-law as the plausible model, the alternative model is still better than the power-law model. We believe that these results are caused by peers that are not willing to maintain large numbers of

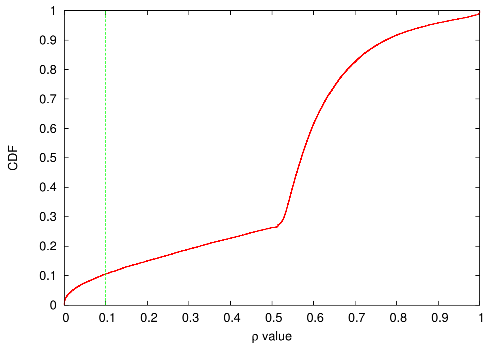

Figure 4.7: CDF plot of ρ value of log-likehood ratio test for power-law v.s log-normal.

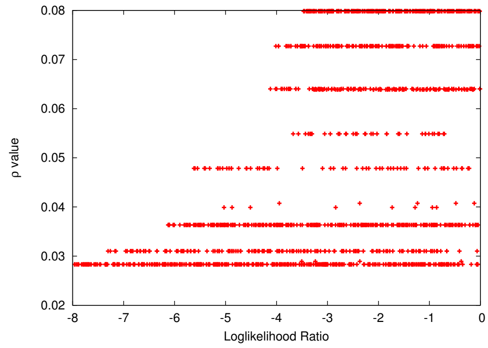

Figure 4.8: Scatter plot p values v.s log-likelihood ratio (LR) for likelihood test for power-law vs log-normal.

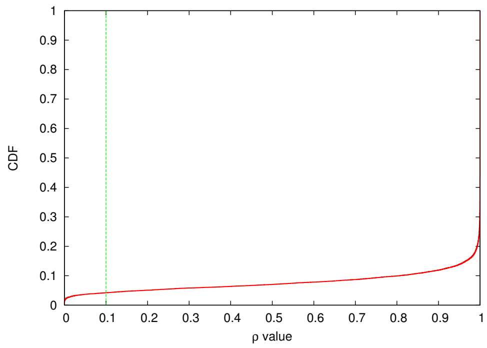

Figure 4.9: CDF plot of ρ value of log-likehood ratio test for power-law v.s exponential.

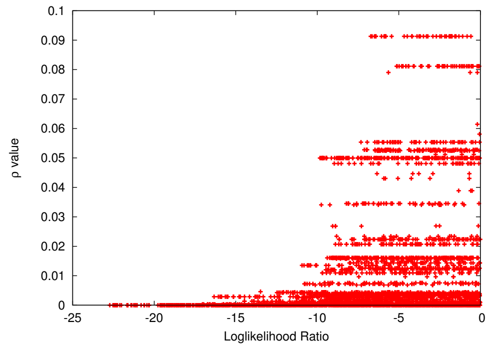

Figure 4.10: Scatter plot p values v.s log-likelihood ratio (LR) for likelihood test for power-law vs exponential.

concurrent connections (high node degree). These observations clearly demonstrate that comparing models is a very complex task in highly dynamic networks.

Non-Nested Case

In the non-nested case, we compare power-law distribution with log-normal distribution and exponential distribution. We calculate the ratio of two likelihood distributions or the logarithm of the ratio, which is positive or negative depending on which distribution is better. The positive or negative sign of log-likelihood ratio does not definitively indicate which model is the better fit. Vuong's [53] method gives a ρ value that can tell us whether the observed sign of likelihood ratio is statistically significant. If this ρ value is small ($\rho < 0.1$) then the sign of log-likehood ratio is a reliable indicator of which model is the better fit to the data. If the ρ value is large the sign of log-likehood ratio is not reliable and the test does not favor either model over the other.

Figure 4.7 and Fig.4.9 show the CDF of the ρ value for the log-likehood ratio between power-law and log-normal distributions and the log-likehood ratio between power-law and exponential distributions. Both alternative distributions only show very small number of data points that have $\rho < 0.1$, each at around 13% and 5.5%. The log-likelihood ratio of these data points have negative signs as shown in Fig.4.8 and Fig.4.10, therefore the alternative distributions are not better than power-law. With the vast majority of the values for ρ being larger than 0.1, the results of the log-likelihood ratio test are ambiguous. Therefore, it is important to look at theoretical factors or design factors behind the systems to make a sensible judgment. For example: a leecher in a Bittorrent swarm is design to prefer the fastest seeders or leechers instead of high degree seeders or leechers. With this design factor information, we can make a sensible judgment which distributional form is more reasonable.

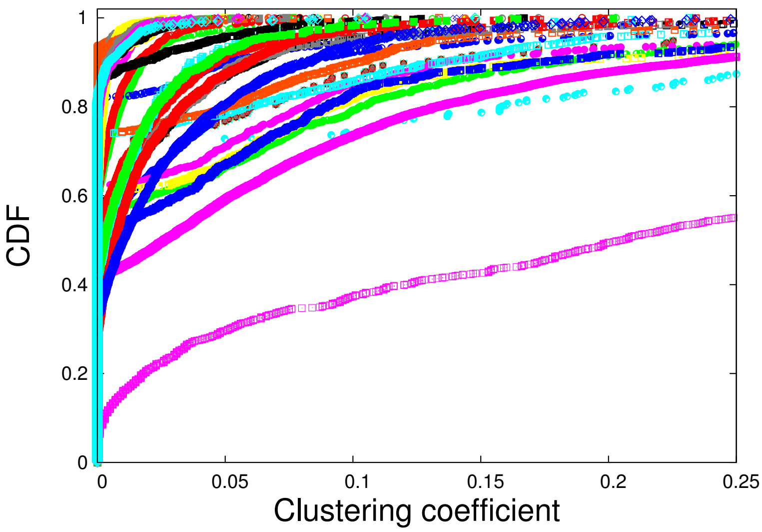

Figure 4.11: CDF plot of the clustering coefficient for each torrent.

4.3.3 Clustering Coefficient

Networks show a tendency for link formation between neighboring vertices called *clustering* that reflects the topology robustness. The clustering around a vertex i is quantified by the clustering coefficient C_i, defined as the number of triangles in which vertex i participates normalized by the maximum possible number of such triangles,

$$c_i = \frac{2t_i}{k_i(k_i-1)} \tag{4.5}$$

where t_i denotes the number of triangles around i and k_i denotes vertex degree. For the whole graph, the clustering coefficient is

$$C = \frac{1}{n}\sum_{i\in G} c_i. \tag{4.6}$$

A larger clustering coefficient represents more clustering at nodes in the graph, therefore the clustering coefficient expresses the local robustness of the network. The distinction between a random and a non-random graph can be measured by clustering-coefficient metrics [55]. A network that has a high clustering coefficient and a small average path length is called a *small-world* model [55]. In Bittorrent systems, a previous study [25] mentioned the possibility that Bittorrent's efficiency partly comes from the clustering of peers. Figure 4.11 shows the CDF clustering coefficient value of our data sets. Only one torrent exhibits clustering coefficient less than 0.1 for about 40% of the snapshots, while for the other torrents, more than 70% are less than 0.1. This low clustering coefficient observation is the same as that observed by Dale *et al*. [9]. Considering only the low clustering coefficient, the Bittorrent topologies seem to be close to random graphs.

4.4 Confirmation via Simulation

We use simulations to compare the overlay topology properties based on our real-world experiments. We set the maximum peer set size to 80, the minimum number of neighbors to 20, and the maximum number of outgoing connections to 80. In our simulation, the results are quite easy to get since we are on a controlled system; we can directly read the global topology properties from our results. We also have the simulated PEX messages. We compare the global overlay topology properties as the final result from the simulator with the overlay topology that we get from PEX on the same simulator. Figure 4.12 shows the α estimate Eq.(4.1) and p Eq.(4.1) value

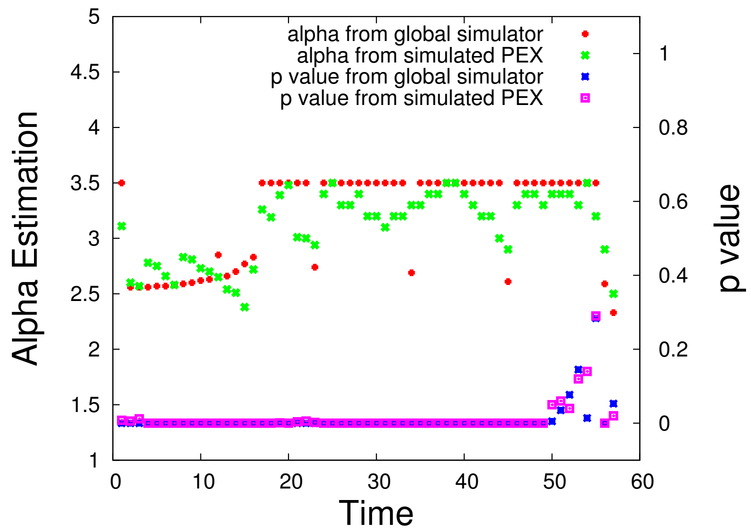

Figure 4.12: α estimation and p value for global topology and topology inferred from PEX where both done in our simulator.

both for the global result and the PEX result from our simulator. It clearly shows that both the global result and the PEX result from the simulator produce very low p values. We calculate the Spearman correlation for both α values from the global result and the PEX result. The Spearman rank correlation coefficient is a non-parametric correlation measure that assesses the relationship between two variables without making any assumptions of a monotonic function. The Spearman rank correlation test gives $0.38 \leq \rho \leq 0.5$, which we consider to be moderately well correlated. Therefore, the simulator confirms that the PEX method can be used to estimate α.

4.5 Related Work

Bittorrent protocol performance has been explored extensively [16] [26] [37] [47] [27] [59]. Although we know that the topology can have a large impact on performance, to date only a few papers have addressed the issue. Urvoy *et al.* [48] used a discrete event simulator to show that the time to distribute a file in a Bittorrent swarm has a strong relation to the overlay topology. Al-Hamra *et al.* [1], also using a discrete event simulator, showed that Bittorrent creates a robust overlay topology and the overlay topology formed is not random. They also show that peer exchange (PEX) generates a chain-like overlay with a large diameter. Dale *et al.* [9], in an experimental study on PlanetLab, show that in the initial stage of Bittorrent a peer will get a random peer list from the tracker. They found that a network of peers that unchoked each other is scale-free and the node degree follows a power-law distribution with exponent approximately 2. Dale *et al.* [9] also showed that the path length formed in Bittorrent swarms averages four hops and Bittorrent swarms have low average clustering coefficient. However, little work has been done on confirming that such controlled experiments correspond to the system.

We emphasize that compared to Hoßfeld *et al.* [18], our work provides a completely different approach and goal. Hoßfeld *et al.* [18] discuss the AS (Autonomous System) level topology of Bittorrent swarm for optimizing overlay traffic across ASes, while our study focus on microscopic dynamic aspect which is Bittorrent swarms topology itself (peer level or IP address level). The closest work to ours is Kryczka *et al.* [24]. While our method is somewhat similar to theirs, they focus on clustering and locality while our focus is on node degree and clustering. They use PEX to discover peer relationship, unfortunately they do not explain in detail how to process the PEX data set. Because of differing PEX implementations between Bittorrent clients, we need to be careful with it in data processing. In our work, we describe PEX behavior and its limitation on two popular Bittorrent clients: Vuze and uTorrent, and we also explain how to treat PEX data from different

Bittorrent clients. We also provide simulation result to confirm that our methods for infering peer relationship with PEX is valid. They observed that Bittorrent swarms have slightly higher clustering coefficient compared to random graphs of the same size and they observe neither Bittorrent swarm fulfills the properties of small world. The slightly difference in clustering come from the difference of PEX data processing. They assume that PEX is the same and complete for all Bittorrent clients, therefore they get slightly different results.

Our results agree with previous research [9] in some areas and disagree in others, perhaps for two reasons. First, power-law claims must be handled carefully. Many steps are required to confirm the power-law behavior, including alternative model checking, and we must be prepared for disappointment since other models may give a better fit. Second, our methodology relies on real work measurement combined with simulation for validation. This real-world measurement will reflect different types of clients connected to our swarm, and each client has a different behavior. We also face difficult-to-characterize network realities such as NAT and firewalls. Our ability to reproduce key aspects of the topology dynamics suggests that these factors have only limited impact on the topology, somewhat to our surprise.

4.6 Summary

We have investigated the properties of Bittorrent overlay topologies from the point of view of the peer exchange protocol using real swarms from an operational Bittorrent tracker on the Internet.

We find that the node degree of the graph formed in a Bittorrent swarm can be described by a power law with exponential cut-off and the observation of a low clustering coefficient implies Bittorrent networks are close to random networks. From the Bittorrent protocol point of view, the reason that a Bittorrent swarm can be described by a power-law with exponential cut-off is: leechers in a Bittorrent swarm prefer a few good seeders or neighbors that can give high data rates to exchange the data and seeders have rich connections to leechers as seeders have complete chunks or pieces. That behavior explains why seeders have rich connections while leechers only have a few neighbors. We argue that there are two reasons for the cut-off phenomenon. First, most Bittorrent clients configure the maximum number of global connection between $200-300$, however the maximum connection per torrent (swarm) is set between $50-90$ by default [39] [50]. Some Bitorrent forums suggest decreasing the maximum connection for torrent (swarm) to between $30-40$ [49]. Second, most of the Bittorrent users are home users where their home gateway device cannot give high concurrent connections and Bitorrent is not the main online activity. We argue that the Bittorrent swarm closes to random that we infer from clustering coeficient is caused by Bittorrent mechanism itself that always choose random peers from its neighbors in the choking-unchoking algorithm, optimistic choking algorithm, and optimistic connect algorithm as we explained previously. Our approach can infer BitTorrent swarms topology and the result confirmed by simulation.

Chapter 5

Green Networking

Since the seminal paper by Gupta and Singh [17], presented at SIGCOMM in 2003, the subject of green networking has received considerable attention. In recent years, valuable efforts have been devoted to reducing unnecessary energy expenditure. Big companies such as Google, Microsoft, and Amazon, are turning to a host of new technologies to reduce operating costs and consume less energy. Google, for example, is planning to operate its data centers with a zero carbon footprint by using, among other things, hydropower, water-based chillers, and external cold air to do some of the cooling. Several approaches have been considered to reduce energy consumption in networks. These include:

- The design of low power components that are still able to offer acceptable levels of performance. For example, at the circuit level techniques such as Dynamic Voltage Scaling (DVS) and Dynamic Frequency Scaling (DFS) can be used. With DVS the supply voltage is reduced when not needed, which results in slower operation of the circuitry. DFS reduces the number of processor instructions in a given amount of time, thus reducing performance. These techniques can reduce energy consumption significantly.
- Consuming energy from renewable energy sources sites rather than incurring in electricity transmission overheads, thus reducing CO_2 emissions.
- Designing new network architectures, for example by moving network equipment and network functions to strategic places. Examples include placing optical amplifiers at the most convenient locations and performing complex switching and routing functions near renewable sources.
- Using innovative cooling techniques. Researchers in Finland, for instance, are running servers outside in Finnish winter, with air temperatures below 20 degrees celsius.
- Performing resource consolidation, capitalizing on available energy. This can be done via traffic engineering, for instance. By aggregating traffic flows over a subset of the network devices and links allows others to be switched off temporarily or be placed in sleep mode. Another way is by migrating computation, typically using virtualization to move workloads transparently.

Chapter 6

Energy Saving in Peer-Assisted CDN

6.1 Motivation

Streaming content, especially video, represents a significant fraction of the traffic volume on the Internet, and it has become a standard practice to deliver this type of content using Content Delivery Networks (CDNs) such as Akamai and Limelight for better scaling and quality of experience for the end users. For example, YouTube uses Google cache and MTV uses Akamai in their operations.

With the spread of broadband Internet access at a reasonable flat monthly rate, users are connected to the Internet 24 hours a day and they can download and share multimedia content. P2P (peer to peer) applications are also widely deployed. In China, P2P is very popular; we see many P2P applications from China such as PPLive, PPStream, UUSe, Xunlei, etc. [52]. Some news broadcasters also rely on P2P technology to deliver popular live events. For example, CNN uses the Octoshape solution that enables their broadcast to scale and offer good video quality as the number of users increases [35].

From the Internet provider point of view, the presence of so many always-on users suggests that it is possible to delegate a portion of computing, storage and networking tasks to the users, thus creating P2P networks where users can share files and multimedia content. Starting from file sharing protocols, P2P architectures have evolved toward video on demand and support for live events.

Broadband network access helps P2P applications to perform better. xDSL networks are deployed worldwide, and in some countries, such as Japan, even higher bandwidth fiber to the home (FTTH) already exceeds DSL in market penetration. In the coming years, network operators throughout the world will massively deploy FTTH. As access bandwidth increases, P2P systems may become more efficient since a peer can contribute much more.

In Peer assisted CDN [1], users can download content from CDN nodes from or other users or peers. A user may cache the content after download to serve requests from other users. Due to the complexity of the behavior of peers, the process should be done in the home gateway user where the ISP can control it.

On the other hand, the data center where the CDN server is placed faces costs for powering the data center. The Uptime Institute, a global data center authority, surveyed 1100 data center owners and operators in 2012 and reported that 55% of organizations will increase their financial budget 10% over 2011 [22]. 30% of organizations were expected run out of data center capacity (power, cooling, space, and network) by the end of 2012 [22]. More than 50% of the organizations surveyed reported that saving energy [2] is a major priority. Even in the data centers using the state of art cooling technologies heat dissipation accounts for at least 20% and as much as 50% of the total power consumption [13]. The increases in energy cost and the demand due to growth of traffic urges the data center operators and owners to look for ways to reduce energy usage in the years to come. Although reducing energy consumption can effectively reduce overall cost, this will limit the capacity for growth and scalability of the service provisioning. For example: routers and servers spend most

[1] In this paper we use Peer Assisted CDN and CDN-P2P interchangeably.

[2] As we are discussing steady-state operation, energy and power are in direct correspondence so we use the terms interchangeably.

of their energy on the baseline activities such as running the fans, spinning disk, powering the backplane, and powering the memory. Even in an idle state, modern systems can be consuming anything from 50% to 80% of the power consumed under maximum load [5, 6].

Alternatively, the data center can be revamped by relocating some services to end-host computers or peers. Peers contribute their communication, storage, and computation resources to exchange data and provide services while the data center performs central administration and authentication as well as backend processing. A P2P network formed by peers offers flexibility and scalability in service delivery.

We study the energy consumption of hybrid CDN-P2P in two use cases: live streaming and online storage services. It has been shown that CDN energy consumption is better than P2P architecture [4, 11]. The questions are: with the opportunity to offload the CDNs workload to the peers, how much energy saving can the CDN server get and how large is the difference compared to a pure CDN architecture. If we can estimate the difference between a CDN architecture and a peer-assisted CDN combined with an estimate of peer power consumption, we can use this difference as a basis calculation for giving an incentive to users since peer assisted relies heavily on the users uptime and upload rate. Furthermore, since the power consumption is reduced, the power requirement inside the data center can be reduced thus relaxing capacity planning.

6.2 System Description

6.2.1 Live Streaming

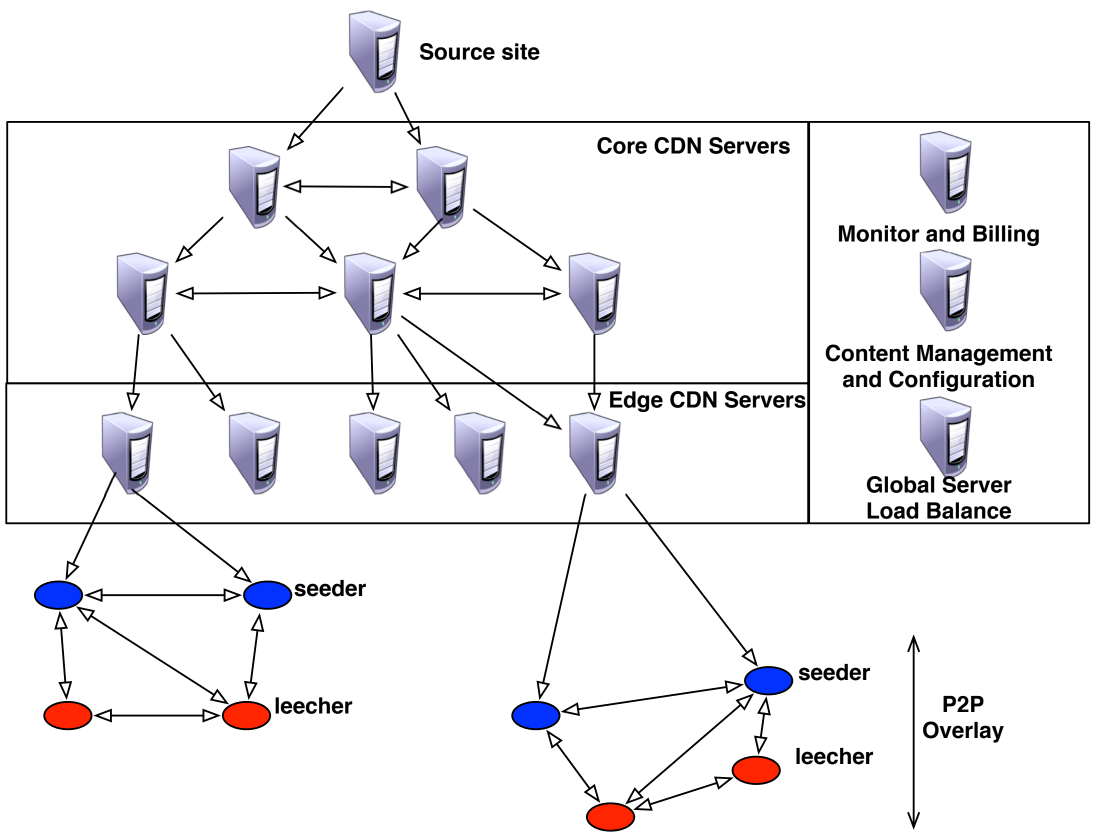

Figure 6.1: Example model of peer-assisted online storage architecture.

Figure 6.1 shows an example model of a peer-assisted CDN for live streaming adapted from [58]. CDN servers deliver video contents from the content provider to end-users. The CDN usually is organized into several tiers usually to cope with the scale demand. Edge CDN servers are directly responsible for serving end users. The goal of the server side peer is for efficient data distribution with some measures to guard against some node failures and network delay.

The CDN overlay is largely tree-based. To provide greater reliability, a CDN node may allow retrieving the content either from other nodes. Edge CDN servers are responsible for serving end users

For this system, we introduce the concepts of seeder and leecher. A peer that is served by an edge CDN server is called a seeder while a peer that is served by seeders is called a leecher.

A peer obtains the URL from a content source. The global server load balancer finds a suitable edge CDN node for this peer. The peer is then redirected to the nearest edge CDN. The edge CDNs has decision logic that decides if a new peer should be served directly by the edge CDN or if it should be redirected to the P2P overlay.

In the P2P overlay, the stream is separated into several substreams according the stream id and peers are organized in a tree-based overlay. A working peer-assisted CDN live streaming system is defined by parameters such as: (1) video bitrate, (2) the total number of peers, (3) the edge CDN servers bandwidth, and (4) peer upload bandwidth capacity and churn rates.

The maximum number of seeders is bounded by the CDN's capacity, while the maximum number of leechers is bounded by the number of seeders with a certain upload rate. Let us denote the number of the seeders by n_s, the number of leechers by n_l, the maximum bitrate supplied by seeders to leechers by ρ, and the video bitrate by r. The number of leechers that can be supported by seeders is:

$$\lfloor n_l \rfloor = n_s.\rho \tag{6.1}$$

The number of seeders that support or upload content to leechers is:

$$n_s^u = n_l.\frac{r}{\rho} \tag{6.2}$$

In peer-assisted live streaming, we introduce the utilization policy where the CDN server admits peers as seeders as long as the CDN utilization does not exceed 50%, which we defined as 50% of the capacity of a Gigabit Ethernet. When the utilization hits 50%, incoming peers are admitted as leechers, hence they receive the contents from seeders. When more peers join the system and the upload capacity of the seeders is exceeded, the policy raises the utilization cap and the server admits the newly joined peers as seeders. We consider this policy to be better than adding a new server from the point of view of energy consumption.

6.2.2 Peer-Assisted Online Storage

Figure 6.2 illustrates the architecture of peer-assisted online storage for a file hosting system (one-click hosting service with peer-assistance) and interactions among the main components [44]. In this system, each file provided by users is treated as a swarm. Each end user is treated as a peer.

In Fig.6.2, arrows 1, 2, and 3 denote the interaction between a participating peer and tracking server and replication servers for uploading a new file. Arrows 4, 5, and 6 denote the interaction between peers and the tracking server to maintain the peer topology. Arrow 7 denotes the sharing of the file and exchange of availability data among peers. Arrows 8 and 9 represent peer requests and server response.

The tracking servers function is to maintain swarm information and bootstraps peers. Replication servers working as dedicated content servers have a function for maintaining the availability of swarms when peers do not actively serve them alone.

We choose this peer-assisted online storage model because this model has been implemented widely in China, e.g. FS2You [12], and because one-click file hosting services are very popular right now [29]. Such services rely heavily on server farms inside the data center, thus energy cost becomes important [3]. In this model, since the server holds an important role in this system, we present a simple mathematical model of server bandwidth allocation strategies as a basis for energy calculations [43,46], as follows:

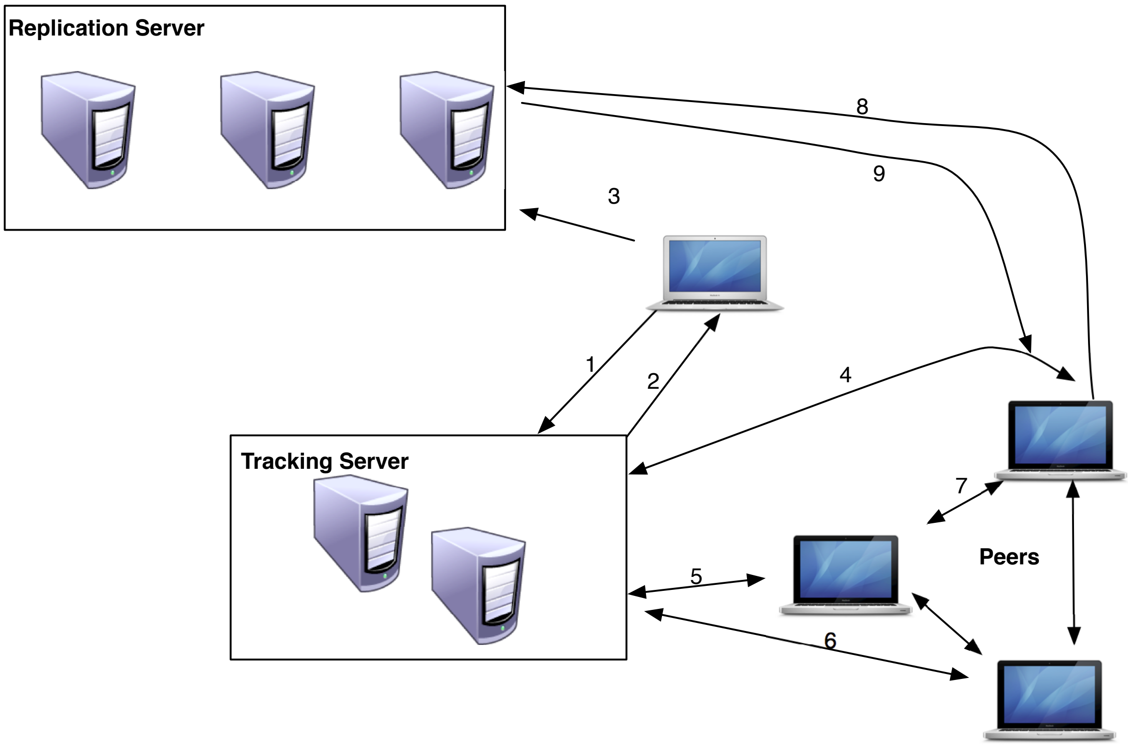

Figure 6.2: Example model of peer-assisted online storage architecture.

- Type-1 represents less popular files and type-2 represents popular files.
- S_{t1} represents server bandwidth allocated to a type-1 file and S_{t2} represents server bandwidth allocated to a type-2 file.
- S is the total server bandwidth.
- S_{max1} is the maximum amount of server bandwidth that can be assigned to a file of type-1 and S_{max2} is the maximum amount of server bandwidth that can be assigned to a file of type-2.
- M_{t1} is the number of type-1 files and M_{t2} is the number of type-2 files.
- μ is upload rate of a peer.
- α_{t1} is the arrival rate of new peers in type-1 file and α_{t2} is the arrival rate of new peers in type-2 file.
- $\alpha = M_{t1}\alpha_{t1} + M_{t2}\alpha_{t2}$
- $M = M_{t1} + M_{t2}$
- η_{t1} is the file sharing effectiveness. It is the fraction of the upload capacity of peers that is being utilized for type-1 file.
- η_{t2} is the file sharing effectiveness. It is the fraction of the upload capacity of peers that is being utilized for type-2 file.
- T_d is the average downloading time.
- x_i is the average number of peers.

There are three server bandwidth allocation strategies: (1) lower bound of the average downloading time; (2) request driven strategy; (3) water leveling strategy. In the lower bound strategy, the server uses the bandwidth for type-1 files until S_{t1} reaches its maximum value, then the residual server bandwidth can be assigned to type-2 files. In the request driven strategy, the server serves every request from peers. The server bandwidth is equally divided among all the peers. Lets assume that the number of requests for a file to the server is proportional to the peer arrival rate of the file. Lets also assume that when the amount of server bandwidth assigned to one of the two types of files has reached its maximum value, the residual server bandwidth will be assigned to the other type of file. In the water leveling strategy, the server bandwidth is equalized across all the files by taking file popularity into consideration. The server serves the requests from peers according to a certain probability, which is inversely proportional to the peer arrival rate of the file. Lets assume that the number of requests for a file to the server is proportional to the peer arrival rate of the file, the server will serve the same number of requests for different files and therefore the server bandwidth is equally allocated across all the files. In order to be able to calculate our power consumption, we need to get the number of peers in the system that can be expressed as [43]:

$$\sum x_i = T_d . \sum \lambda_i \tag{6.3}$$

Furthermore, we can calculate T_d:

$$T_d = \frac{1}{M_{t1}.\lambda_{t1} + M_{t2}.\lambda_{t2}} \left(\frac{M_{t1}.f_{t1}.\lambda_{t1}.\eta_{t2} + M_{t2}.f_{t2}.\lambda_{t2}.\eta_{t1}}{\mu.\eta_{t1}.\eta_{t2}} - \frac{S_{t1}(M_{t1}.\eta_{t2} - M_{t2}.\eta_{t1}) + S.M_{t2}.\eta_{t1}}{\mu.\eta_{t1}.\eta_{t2}} \right) \tag{6.4}$$

6.2.3 Energy Model

Our goal is to provide a general view and fair comparison of the energy consumed by a pure CDN and a hybrid CDN-P2P architecture. To do so, we designed a series of models and performed an analysis. Our energy model is similar to the models used in [33]. The differences with [33] are, firstly, our baseline energy is not a function of bitrate flow. Our baseline energy is based on the minimum energy required to turn on the device without any traffic flowing through the device. Secondly, our overhead ratio is based on the Coefficient of Performance (COP) of the cooling cycle in data center, which we will explain at the end of this section.

Let E_s, E_r, and E_p denote the energy consumption of a single request at each a CDN server, router, and peer respectively. Next, we define baseline energy consumption as the energy consumed to keep the device on, even when there is no traffic. Let E_{s-base}, E_{r-base}, and E_{p-base} denote the baseline energy consumption for CDN server, router, and peer respectively; and E_{s-max}, E_{r-max}, E_{p-max} denote the power consumption of server, router, and peer when operating at the maximum capacity.

Next, we introduce work-induced energy as the energy consume per bit transferred. Let δ_s,δ_r, and δ_p denote the work-induced energy consumed per additional bit transferred by each CDN server, router, and peer,

$$\delta_s = \frac{(E_{s-max} - E_{s-base})}{M_s} \tag{6.5}$$

$$\delta_r = \frac{(E_{r-max} - E_{r-base})}{M_r} \tag{6.6}$$

$$\delta_p = \frac{(E_{p-max} - E_{p-base})}{M_p} \tag{6.7}$$

Furthermore, we can get:

$$E_s = \delta_s B + E_{s-base} \tag{6.8}$$

$$E_r = d\delta_r B + E_{r-base} \tag{6.9}$$

$$E_p = \delta_p B + E_{p-base} \tag{6.10}$$

where d is the number of hops and B is the size of file to be transferred in bits.

We now introduce the overhead for the server and routers. The only overhead that we will consider here is cooling power. Since servers and routers are placed in the data center, the data center needs to be provisioned with adequate cooling. This cooling overhead in the data center is quantified by the coefficient of performance (COP). The COP value itself has been empirically determined to be [32]:

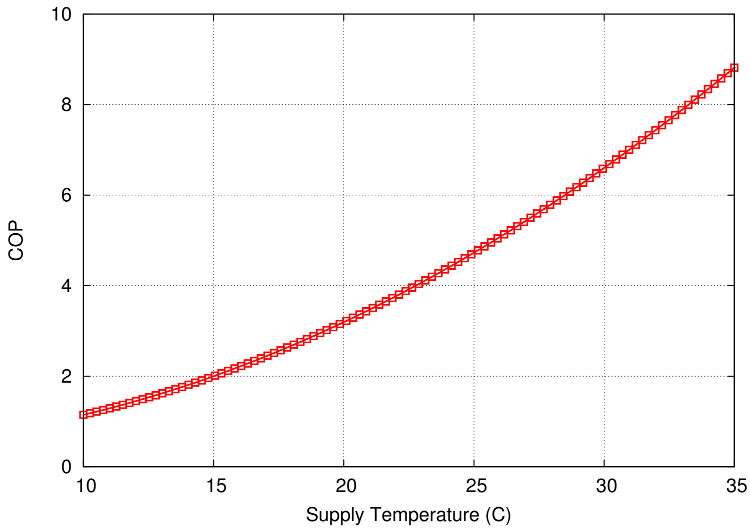

Figure 6.3: COP curve for the chilled water cooling units from HP Lab utility data center. As the target temperature of the air the cooling unit pumps into the floor plenum increases, the COP increases.

$$COP(T) = 0.0068.T^2 + 0.0008.T + 0.458 \tag{6.11}$$

Where T is the temperature supplied by the cooling unit in Celsius. Figure 6.3 shows the $COP(T)$ value for every T. Finally, the cooling cost can be calculated as [32]:

$$C = \frac{Q}{COP(T)} \tag{6.12}$$

Where Q is the amount of power consumed by the servers and hardware. We assume a uniform T at each cooling unit. Taking into account the cooling energy overhead, the total energy consumption is as follows:

$$E_t = E_s\left(1+\frac{1}{COP(T)}\right)+E_r\left(1+\frac{1}{COP(T)}\right) \tag{6.13}$$

We do not include the cooling overhead in the peer energy consumption because most of the peers in homes do not need a separate cooling supply.

6.3 Result and Analysis

6.3.1 Numerical Parameters

The parameters used in this analysis were adapted from [4, 33, 43, 51]. The parameters values are shown in Table 6.1. We choose the numerical parameters from [4, 33, 43, 51] because these parameters were gathered from empirical measurements.

Table 6.1: Numerical Simulation Parameters.

Symbol	Description	Values
δ_s	Work induced at server per bit transferred	$5.2.10^{-8}$ (J/b)
δ_r	Work induced at router per bit transferred	$8.0.10^{-9}$ (J/b)
δ_p	Work induced at peer per bit transferred	$16.10^{-}9$ (J.b)
E_{r-base}	Router baseline power consumption	750 watt
E_{s-base}	Server baseline power consumption	290 watt
E_{p-base}	Peer baseline power consumption	13.5 watt
r	Video bitrate in live streaming	1Mbps
d	Number of hops	1
N_s^u	Upload rate of peers in live streaming	$[0.25, 0.5, 0.75, 1]$ Mbps
N	Number of peers in live streaming	$[100, .., 1000]$
δ_{t1}	Type-1 peer arrival rate to less popular files in online storage (Poisson process)	0.1
δ_{t2}	Type-2 peer arrival rate to less popular files in online storage (Poisson process)	1
η_{t1}	File type-1 sharing effectiveness. The fraction of the upload capacity of peers that is being utilized in online storage	0.5
η_{t2}	File type-2 sharing effectiveness. The fraction of the upload capacity of peers that is being utilized in online storage	1
M_{t1}	Number of files in type-1 files or less popular files	10
M_{t2}	Number of files in type-2 files or less popular files	1
$f_{t1} = f_{t2}$	File size in online storage	100 MB
μ	Upload rate of peers in online storage	0.5Mbps
c	Downloading rate of peers in online storage	1Mbps
T	Air temperature supplied form cooling unit in data center	$[20, 25]$ correspond to COP value $[3.194, 4.728]$

6.3.2 Live Streaming Service

In the live streaming case, each video stream flows from the CDN node to the network, in this case a router, and then arrives at the seeders. If seeders need to upload data to leechers it will flow through the router then arrive at the leechers. We show the logical network architecture of peer-assisted CDN for live streaming in Fig.6.1. Fig.6.4 shows a simplified physical representation of the network. While in a logical network the peer can communicate directly with another peer, in a real physical world the communication between peers always passes through a router inside the data center.

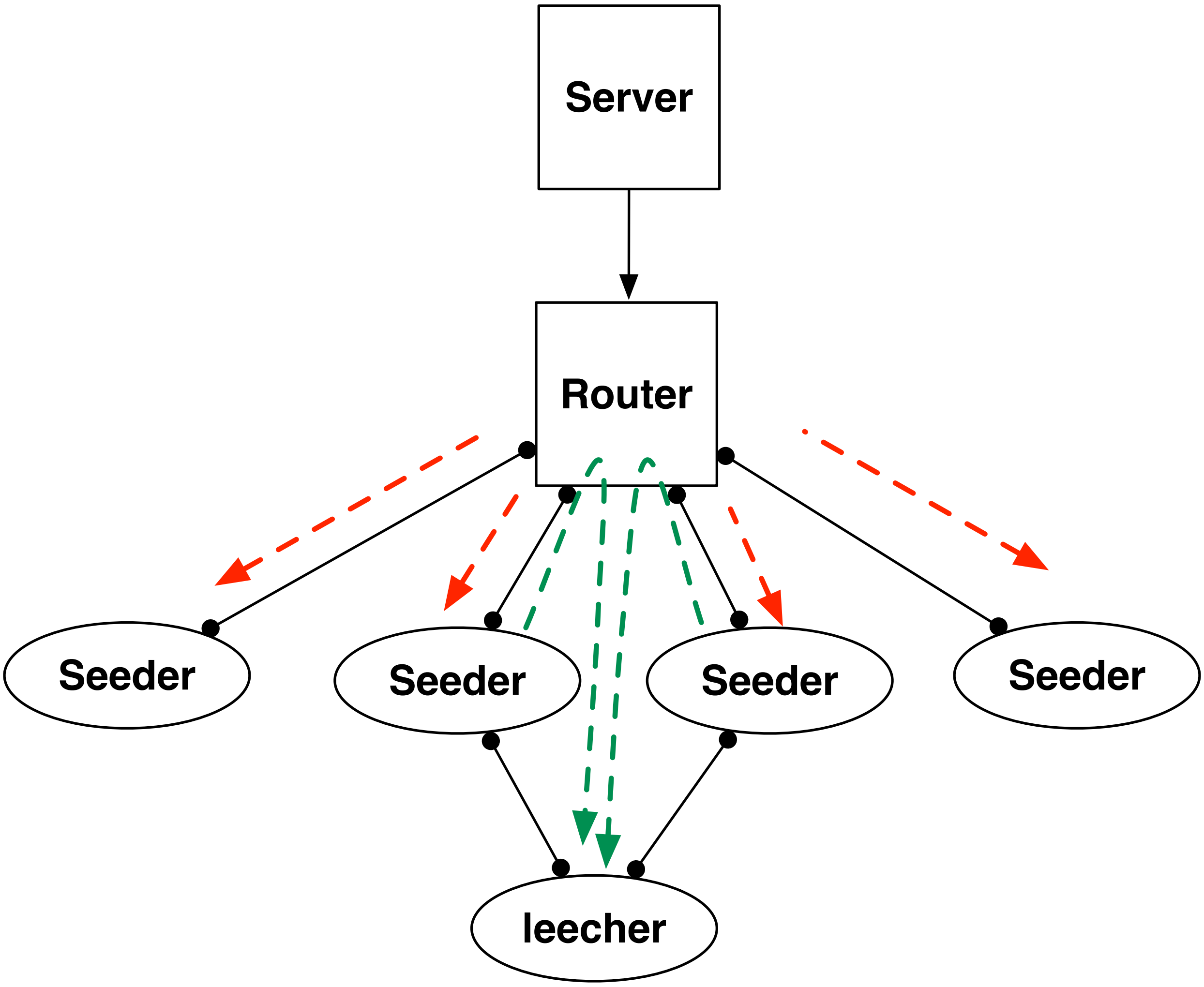

Figure 6.4: Simplified physical representation flow of data.

Figure 6.5 shows energy usage for CDN server, router, and the total energy consumption for the CDN scenario (without peer assist). We plot the energy consumption for CDN server, router, clients, and total energy for two COP coefficient values (T). All energy consumption components increase in value as the number of peers increases; and peers consume most of the energy. The effect of variations in T on total energy is small

Figure 6.6 shows the energy consumption of all components for the CDN-P2P scenario. We use the peer upload rate $N_s^u = 0.75$ in Fig.6.6. We observe that there is almost no change in peers power consumption compared to the pure CDN. The router consumes more power with a higher rate of increase because in CDN-P2P peers the traffic originated from the seeders passes trough the router twice. The CDN server power consumption has small increases between $N = 100$ and $N = 500$, while there are sections with no power increase at $N > 500$, which is where the seeders are uploading contents to the leechers. The server power consumption remains flat as long as the upload rate does not exceed the defined peer upload rate.

Figure 6.7 shows the energy savings of CDN-P2P compared to CDN architecture for CDN server with the utilization policy as explained in Sec.6.2 for $N_s^u = [0.25, 0.5, 0.75, 1]$.

Lets take $N_s^u = 0.75$ as an example. The first 500 nodes are served directly by the CDN server since the utilization of the CDN server is 50% or less. We consider 500 nodes to be 50% utilization because a video-rate of 1 Mbps will result in total network traffic of 0.5 Gbps, which is half of a Gigabit Ethernet interface. When more peers join the system, these peers will be treated as leechers as long as the upload ratio condition is fulfilled. The number of leechers that can be supported by seeders is 375. Therefore from $N = 500$ to $N = 875$, the CDN server does not need to increase utilization because the leechers can be supported by seeders, thus we

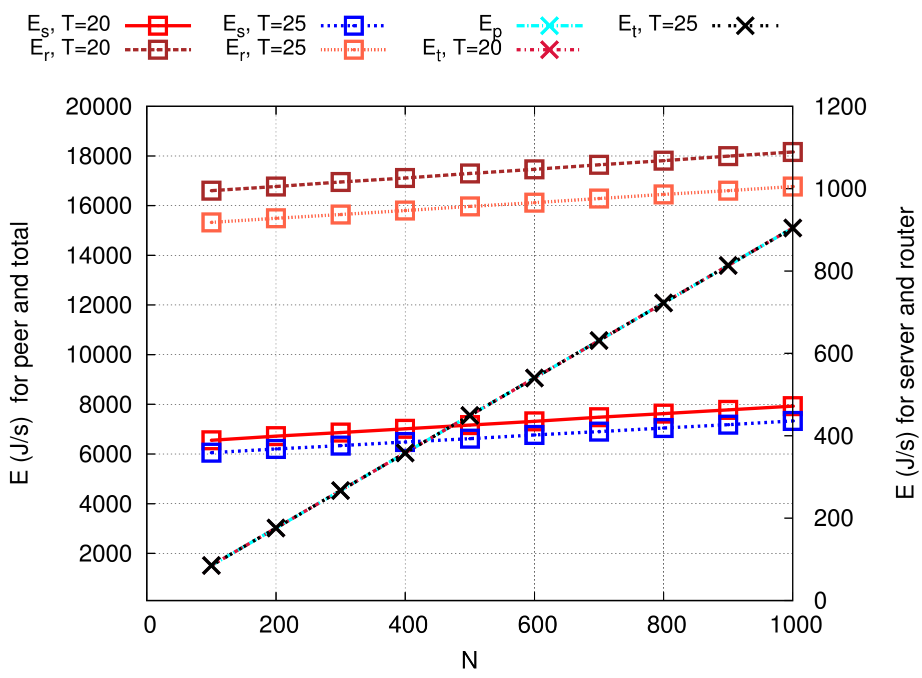

Figure 6.5: Power consumption for the server, router, peer, and total system for CDN architecture. Note that the server and router energy are plotted using the right hand scale, and the peer and total energy are plotted using the left hand scale.

see that the CDN server saves energy. In this phase, compared to CDN architecture, CDN-P2P energy saving is around 7.8%. Next, we have 875 total peers in the system, is apportioned into 500 peers as seeders and 375 as leechers. Since more peers joining the system, the CDN increases the utilization from 50% to 87.5% so all current 875 peers become seeders. In this phase, 875 seeders can support an additional 656 leechers. Therefore, from $N = 875$ to $N = 1531$ the CDN utilization is flat at 87.5% because 875 seeders can support 656 leechers thus we have energy savings around 11% compared to CDN architecture. Other values of N_s^u have same pattern as shown in Fig.6.7.

Figure 6.8 shows the total energy savings of CDN-P2P compared to CDN architecture for $N_s^u = [0.25, 0.5, 0.75, 1]$. As the savings only occurs in the CDN server, we see the same patterns as in Fig 6.7 but with a much lower percentage of energy savings, which is 1%.

6.3.3 Online Storage

To calculate the energy consumption in peer-assisted online storage, we must be able to determine the number of peers in the system. We get average downloading time (T_d) values by varying server bandwidth values (S) from 0 to 150 MBps using Eq 6.4. After getting the average downloading time, we can get the average number of peers using Eq 6.3. We found that the number of peers is inversely related to the server bandwidth. The number of peers is the horizontal axis in Fig 6.9 and Fig 6.10. The figures cover more number of peers compared to the live streaming service and we can look at the comparable number of peers in both cases when we want to do comparisons.

Figure 6.9 shows the power consumption for the lower-bound strategy for the server, router, peers, and the total system. We found that increasing the number of peers decreases CDN server power consumption because the bandwidth usage of the CDN server decreases. The router power consumption is flat at around 1000J/s because the server bandwidth reduction is offset by the increasing number of peers. We also found that the other strategies, request driven and water leveling, have the same pattern as the power consumption of the lower-bound strategy.

Figure 6.10 shows a comparison of the energy consumption between the request driven and the lower bound strategy, and between the water-leveling strategy and the lower bound strategy for different numbers

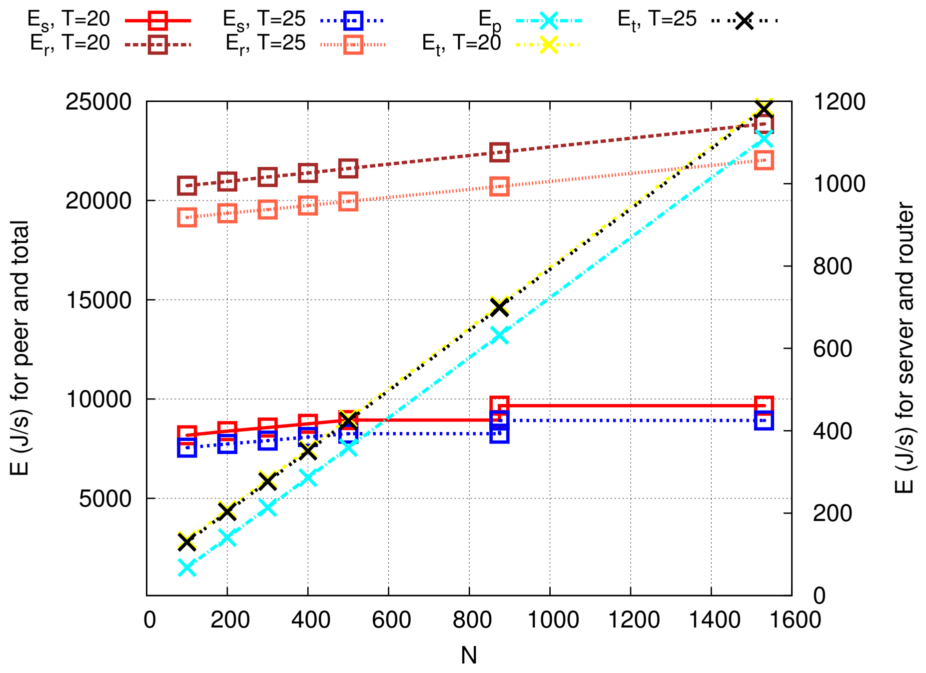

Figure 6.6: Power consumption for the server, router, peer and total system for peer-assisted CDN with N_s^u = 0.75. Note that the server and router energy are plotted using the right hand scale, and the peer and total energy are plotted using the left hand scale.

of peers. Compared to the water-leveling strategy, the request driven strategy required more energy because the request driven strategy equalizes the server bandwidth across all the peers. The water-leveling strategy equalizes server bandwidth across all the files by taking file popularity into consideration, thus minimizing downloading time. We mentioned before that the number of peers is inversely related to the server bandwidth, therefore for the same server bandwidth, we get different numbers of peers for each strategy. This implies that for the same number of peers, we get different server bandwidth. That is the reason for $1000 < N < 2500$ the power consumption diverges. In very limited server bandwidth (less than 45MBps) and sufficient server bandwidth (more than 120MBps) each strategy has the same downloading performance. That is the reason for $N < 1000$ and $N > 2500$ we have the same the number of peers for same bandwidth. As shown in Fig 6.10, for $N < 1000$ and $N > 2500$ the savings for each strategy is relatively the same.

File popularity has a strong correlation with downloading performance. We examine popularity by varying the peer arrival rate of less popular files while the server has fixed bandwidth. Specifically, we increase the type-1 files popularity from 0.1 to 1 and we choose a fixed server bandwidth $S = 50$ MBps which is similar to FS2You. Figure 6.11 shows the difference in total power consumption (left axis) and router power consumption (right axis) of that case. Since the server bandwidth is fixed, we only show the power consumption changes in the routers and the total. When the arrival rate is less than 0.5, the request driven strategy has worse downloading performance compared to the water leveling strategy. This implies more peers exist in the request driven strategy than the water leveling strategy. Therefore, the energy consumption of the request driven strategy is higher than the water leveling strategy. Generally for each strategy, increasing the peer arrival rates to less popular file makes the total energy consumption and router energy consumption increase because more peers are present in the system. Increasing the peer arrival rates to the less popular file makes both the request driven and the water leveling strategy energy consumption converge to a lower bound. This is because more peers in the system improve P2P content availability, thus improving downloading performance that converges to the lower bound strategy.

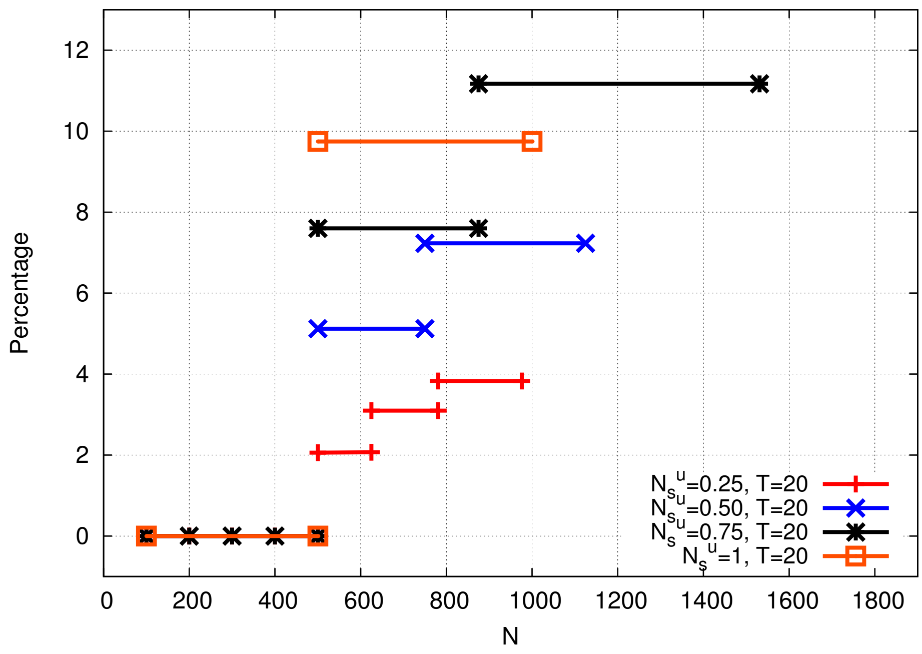

Figure 6.7: Savings in power consumption between CDN architecture and peer-assisted CDN for server with $N_s^u = [0.25, 0.5, 0.75, 1]$.

6.4 Summary

We compared the energy consumption between peer-assisted CDN and pure CDN for live streaming and online storage services both at the data center as well as in total. Employing peer-to-peer capability to assist a CDN is thought to lower the energy requirements at data centers, and we found that the maximum savings at the data centers are 11% and 21%, respectively for the live streaming and online storage services. These savings may change depending on the COP values used and should be better if a new generation of power proportional server were used. One thing to note is that as the number of peers increases, the servers energy consumption increases for the live streaming and decreases for the online storage service due to the differences in the ways both services handle peers. However, the servers energy consumption is swamped by the peers energy consumption. Despite this difference in behavior in the two cases, when comparing Peer Assisted CDN to pure CDN, we found the total energy savings of less than 1%. Nevertheless, the total energy consumption is large, so that even a small percentage improvement results in valuable net reduction. Several areas that we identified as the future work are: 1. The effect of peers uptime variations; 2. More realistic file popularity models for the online storage service; and 3. How CDN providers or ISPs give incentives to the peers based on the understanding of the energy consumptions.

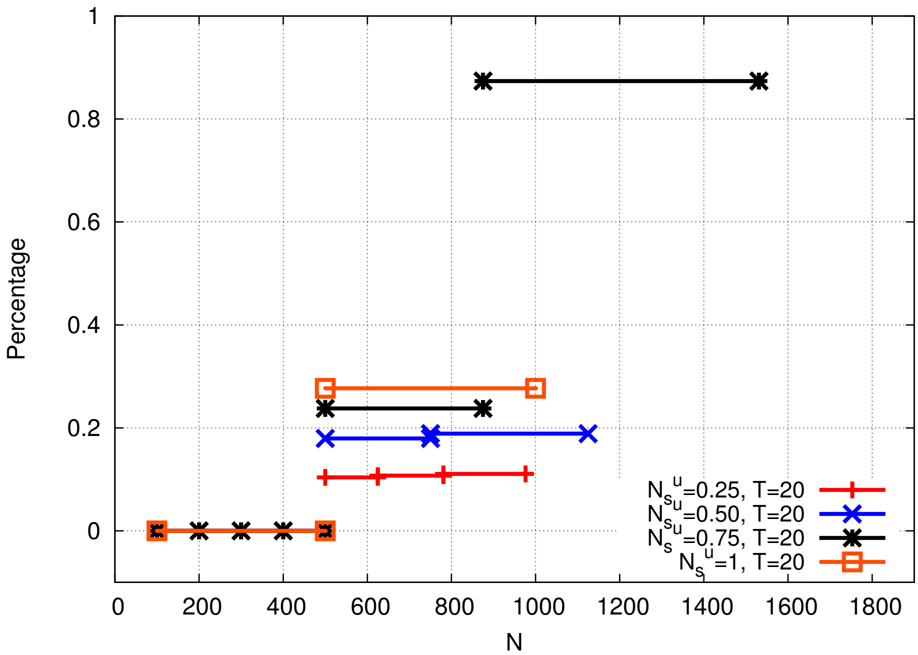

Figure 6.8: Savings in power consumption between CDN architecture and peer-assisted CDN for total system with $N_s^u = [0.25, 0.5, 0.75, 1]$.

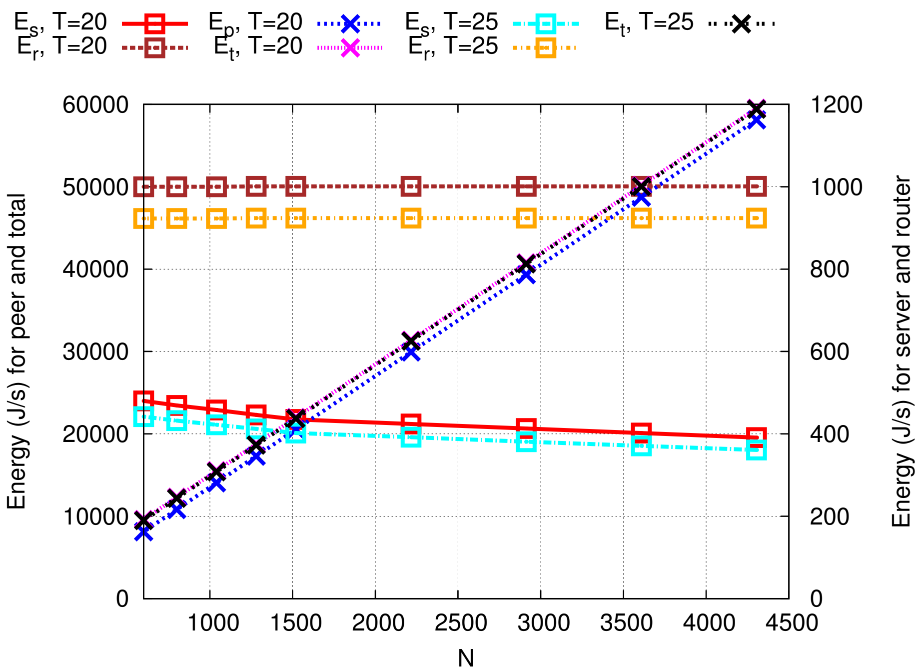

Figure 6.9: Power consumption for lower bound strategy. Note that the server and router energy are plotted using the right hand scale, and the peer and total energy are plotted using the left hand scale.

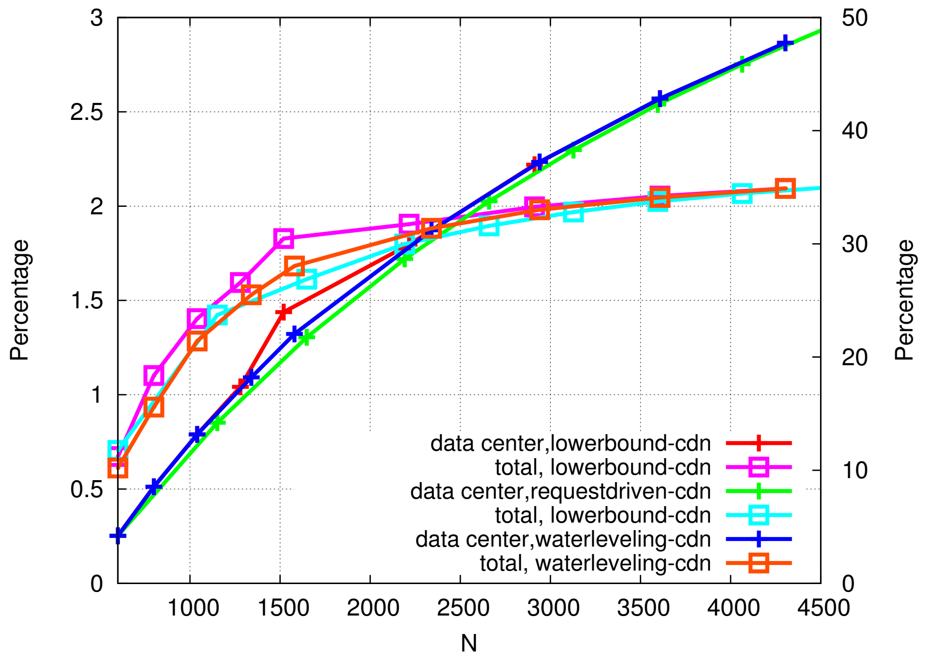

Figure 6.10: Savings in power consumption between each bandwidth allocation and CDN architecture for total and server. Note that the server and router energy (data center) are plotted using the right hand scale, and the peer and total energy are plotted using the left hand scale.

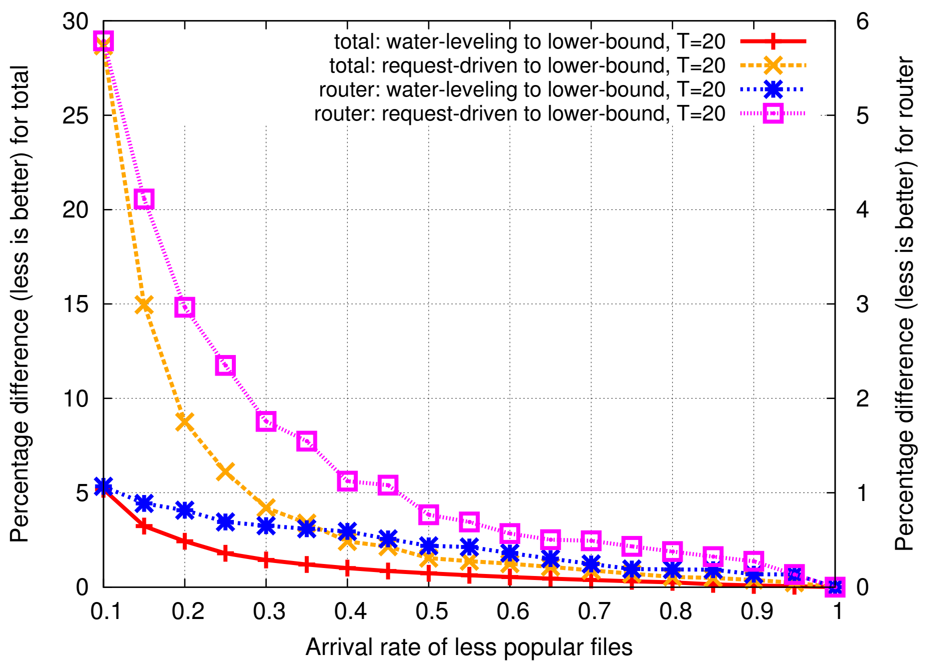

Figure 6.11: Power consumption for total (left axis) and router (right axis) under different server bandwidth allocation strategies when peer arrival rate of less popular varies. We use fixed server bandwidth $S = 50$ MBps.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

We have investigated the properties of Bittorrent overlay topologies from the point of view of the peer exchange protocol using real swarms from an operational Bittorrent tracker on the Internet.

Our results agree in some particulars and disagree in others with prior published work on isolated testbed experiments, suggesting that more work is required to fully model the behavior of real-world Bittorrent networks.

We find that the node degree of the graph formed in a Bittorrent swarm can be described by a power law with exponential cut-off and the observation of a low clustering coefficient implies Bittorrent networks are close to random networks. From the Bittorrent protocol point of view, the reason that a Bittorrent swarm can be described by a power-law with exponential cut-off is: leechers in a Bittorrent swarm prefer a few good seeders or neighbors that can give high data rates to exchange the data and seeders have rich connections to leechers as seeders have complete chunks or pieces. That behavior explains why seeders have rich connections while leechers only have a few neighbors. We argue that there are two reasons for the cut-off phenomenon. First, most Bittorrent clients configure the maximum number of global connection between $200-300$, however the maximum connection per torrent (swarm) is set between $50-90$ by default [39] [50]. Some Bitorrent forums suggest decreasing the maximum connection for torrent (swarm) to between $30-40$ [49]. Second, most of the Bittorrent users are home users where their home gateway device cannot give high concurrent connections and Bitorrent is not the main online activity. We argue that the Bittorrent swarm closes to random that we infer from clustering coeficient is caused by Bittorrent mechanism itself that always choose random peers from its neighbors in the choking-unchoking algorithm, optimistic choking algorithm, and optimistic connect algorithm as we explained previously.

Integrating peer to peer capability to assist the existing CDN has a potential to save energy consumption. In this study, we show that event without explicitly considering energy consumption while assigning content, the peer assisted CDN can save energy consumption. Although the energy savings depend on number of request (number of clients), number of router and its configuration, for total system energy saving is around 0.5 to 1.2. If we break per component, the CDN server is the part that can be push to save energy up to 11% and can be more if new generation of power proportional server is used [23]. We agree with [6], Router component in the other side is quite difficult for energy saving, because different chassis size, different network interface type slot, and different configuration has different energy consumption. Several areas that we have been identified for future work are: more correlation analysis of time period to peer energy usage pattern in live streaming, continued characterization of different peer energy usage based on flash memory storage, and comparing energy model with different file popularity models.

7.2 Future Work

Some areas of improvement that we have identified for future work are: more correlation analysis of the number of peers with α and p value, continued characterization with NATed peers, wider likelihood ratio test with other models and comparing the results with simulation for global graph properties such as distance distribution and spectrum. We hope to incorporate these properties into a complete dK series for the evolution of a real-world Bittorrent overlay as it evolves over time [28]. We conclude that further work throughout the community is necessary to continue to improve the agreement of simulation and controlled experiment with the real world, and that such work will impact our understanding of Bittorrent performance and its effects on the Internet. Several areas that we have been identified for future work in energy comsumption of peer-assisted CDN: more correlation analysis of time period to peer energy usage pattern in live streaming, continued characterization of different peer energy usage based on flash memory storage, and comparing energy model with different file popularity models.

Bibliography

[1] Al Hamra, A., Legout, A., and Barakat, C. Understanding the properties of the bittorrent overlay. *CoRR abs/0707.1820* (2007).

[2] Al-Hamra, A., Liogkas, N., Legout, A., and Barakat, C. Swarming overlay construction strategies. In *Computer Communications and Networks, 2009. ICCCN 2009. Proceedings of 18th Internatonal Conference on* (2009), IEEE, pp. 1–6.

[3] Arstechnica. Megaupload data cost. `http://arstechnica.com/tech-policy/2012/03/isp-storing-25-petabytes-of-megaupload-data-costs-us-9000-a-day/`.

[4] Baliga, J., Hinton, K., and Tucker, R. S. Energy consumption of the internet. In *Optical Internet, 2007 and the 2007 32nd Australian Conference on Optical Fibre Technology. COIN-ACOFT 2007. Joint International Conference on* (2007), IEEE, pp. 1–3.

[5] Barroso, L., and Holzle, U. The case for energy-proportional computing. *Computer 40*, 12 (2007), 33–37.

[6] Chabarek, J., Sommers, J., Barford, P., Estan, C., Tsiang, D., and Wright, S. Power awareness in network design and routing. In *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE* (2008), pp. 457–465.

[7] Cisco. Cisco visual networking index. `http://www.cisco.com/en/US/netsol/ns827/networking_solutions_sub_solution.html`.

[8] Clauset, A., Shalizi, C., and Newman, M. Power-law distributions in empirical data. *SIAM review 51*, 4 (2009), 661–703.

[9] Dale, C., Liu, J., Peters, J., and Li, B. Evolution and enhancement of bittorrent network topologies. In *Quality of Service, 2008. IWQoS 2008. 16th International Workshop on* (2008), IEEE, pp. 1–10.

[10] Doyle, J., Alderson, D., Li, L., Low, S., Roughan, M., Shalunov, S., Tanaka, R., and Willinger, W. The robust yet fragile nature of the internet. *Proceedings of the National Academy of Sciences of the United States of America 102*, 41 (2005), 14497.

[11] Feldmann, A., Gladisch, A., Kind, M., Lange, C., Smaragdakis, G., and Westphal, F.-J. Energy trade-offs among content delivery architectures. In *Telecommunications Internet and Media Techno Economics (CTTE), 2010 9th Conference on* (2010), IEEE, pp. 1–6.

[12] FS2You. Fs2you. `http://www.rayfile.com/en/`.

[13] Google. Google data center power usage. Available on `http://www.google.com/corporate/datacenters/measuring.html`.

[14] Grindrod, P., and Higham, D. J. Evolving graphs: dynamical models, inverse problems and propagation. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Science 466*, 2115 (Mar. 2010), 753–770.

[15] Guan, K., Atkinson, G., Kilper, D., and Gulsen, E. On the energy efficiency of content delivery architectures. In *Communications Workshops (ICC), 2011 IEEE International Conference on* (2011), pp. 1–6.

[16] Guo, L., Chen, S., Xiao, Z., Tan, E., Ding, X., and Zhang, X. Measurements, analysis, and modeling of bittorrent-like systems. In *Proceedings of the 5th ACM SIGCOMM conference on Internet Measurement* (2005), USENIX Association, p. 4.

[17] Gupta, M., and Singh, S. Greening of the internet. In *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications* (New York, NY, USA, 2003), SIGCOMM '03, ACM, pp. 19–26.

[18] Hossfeld, T., Hock, D., Oechsner, S., Lehrieder, F., Despotovic, Z., Kellerer, W., and Michel, M. Measurement of bittorrent swarms and their as topologies. Technical Report 464, University of Wurzburg, November 2009.

[19] Huang, C., Li, J., and Ross, K. Peer-assisted vod: Making internet video distribution cheap. In *Proc. of IPTPS* (2007), vol. 7.

[20] Huang, C., Li, J., and Ross, K. W. Can internet video-on-demand be profitable? *SIGCOMM Comput. Commun. Rev. 37*, 4 (Aug. 2007), 133–144.

[21] Huang, C., Wang, A., Li, J., and Ross, K. W. Understanding hybrid cdn-p2p: why limelight needs its own red swoosh. In *Proceedings of the 18th International Workshop on Network and Operating Systems Support for Digital Audio and Video* (New York, NY, USA, 2008), NOSSDAV '08, ACM, pp. 75–80.

[22] Institute, U. the 2012 uptime institute data center industry survey. Available on `http://uptimeinstitute.com/2012-survey-results`.

[23] Krioukov, A., Mohan, P., Alspaugh, S., Keys, L., Culler, D., and Katz, R. Napsac: design and implementation of a power-proportional web cluster. *SIGCOMM Comput. Commun. Rev. 41*, 1 (Jan. 2011), 102–108.

[24] Kryczka, M., Cuevas, R., Guerrero, C., and Azcorra, A. Unrevealing the structure of live bittorrent swarms: Methodology and analysis. In *Proc. IEEE Int Peer-to-Peer Computing (P2P) Conf* (2011), pp. 230–239.

[25] Legout, A., Liogkas, N., Kohler, E., and Zhang, L. Clustering and sharing incentives in bittorrent systems. In *Proceedings of the 2007 ACM SIGMETRICS international conference on Measurement and modeling of computer systems* (2007), ACM, pp. 301–312.

[26] Legout, A., Urvoy-Keller, G., and Michiardi, P. Rarest first and choke algorithms are enough. In *Proceedings of the 6th ACM SIGCOMM conference on Internet measurement* (2006), ACM, pp. 203–216.

[27] Li, Z., Goyal, A., Chen, Y., and Kuzmanovic, A. Measurement and diagnosis of address misconfigured p2p traffic. In *INFOCOM, 2010 Proceedings IEEE* (2010), IEEE, pp. 1–9.

[28] Mahadevan, P., Krioukov, D., Fall, K., and Vahdat, A. Systematic topology analysis and generation using degree correlations. In *Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications* (2006), ACM, pp. 135–146.

[29] Mahanti, A. Measurement and analysis of cyberlocker services. In *Proceedings of the 20th international conference companion on World wide web* (New York, NY, USA, 2011), WWW '11, ACM, pp. 373–378.

[30] Mandal, U., Lange, C., Gladisch, A., and Mukherjee, B. Should isps adopt hybrid cdn-p2p in ip-over-wdm networks: An energy-efficiency perspective? In *Advanced Networks and Telecommuncations Systems (ANTS), 2012 IEEE International Conference on* (2012), pp. 6–11.

[31] Mitzenmacher, M. A brief history of generative models for power law and lognormal distributions. *Internet mathematics 1*, 2 (2004), 226–251.

[32] Moore, J., Chase, J., Ranganathan, P., and Sharma, R. Making scheduling" cool": temperature-aware workload placement in data centers. In *Proceedings of the annual conference on USENIX Annual Technical Conference* (2005), pp. 5–5.

[33] Nedevschi, S., Ratnasamy, S., and Padhye, J. Hot data centers vs. cool peers. In *Proceedings of the 2008 conference on Power aware computing and systems* (Berkeley, CA, USA, 2008), HotPower'08, USENIX Association, pp. 8–8.

[34] Nordberg, A. Rasterbar libtorrent. `http://www.rasterbar.com/products/libttorrent`.

[35] Octoshape. Cnn uses octoshape. Available on `http://octoshape.com/cnn-com-using-octoshapes-p2p-for-live-feed/`.

[36] Palasamudram, D. S., Sitaraman, R. K., Urgaonkar, B., and Urgaonkar, R. Using batteries to reduce the power costs of internet-scale distributed networks. In *Proceedings of the Third ACM Symposium on Cloud Computing* (New York, NY, USA, 2012), SoCC '12, ACM, pp. 11:1–11:14.

[37] Pouwelse, J., Garbacki, P., Epema, D., and Sips, H. A measurement study of the bittorrent peer-to-peer file-sharing system. *Delft University of Technology Parallel and Distributed Systems Report Series, Tech. Rep. Technical Report PDS-2004-007* (2004).

[38] Qureshi, A., Weber, R., Balakrishnan, H., Guttag, J., and Maggs, B. Cutting the electric bill for internet-scale systems. *ACM SIGCOMM Computer Communication Review 39*, 4 (2009), 123–134.

[39] Setting, V. B. C. G. Available on `http://wiki.vuze.com/w/Good_settings`.

[40] Stutzbach, D., and Rejaie, R. Understanding churn in peer-to-peer networks. In *Proceedings of the 6th ACM SIGCOMM conference on Internet measurement* (2006), ACM, pp. 189–202.

[41] Stutzbach, D., Rejaie, R., Duffield, N., Sen, S., and Willinger, W. Sampling techniques for large, dynamic graphs. In *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings* (2007), IEEE, pp. 1–6.

[42] Stutzbach, D., Rejaie, R., and Sen, S. Characterizing unstructured overlay topologies in modern p2p file-sharing systems. *Networking, IEEE/ACM Transactions on 16*, 2 (2008), 267–280.

[43] Sun, Y., Liu, F., Li, B., and Li, B. Peer-assisted online storage and distribution: modeling and server strategies. In *Proceedings of the 18th international workshop on Network and operating systems support for digital audio and video* (New York, NY, USA, 2009), NOSSDAV '09, ACM, pp. 13–18.

[44] Sun, Y., Liu, F., Li, B., Li, B., and Zhang, X. Fs2you: Peer-assisted semi-persistent online storage at a large scale. In *INFOCOM 2009, IEEE* (2009), pp. 873–881.

[45] Tang, J., Musolesi, M., Mascolo, C., and Latora, V. Temporal distance metrics for social network analysis. In *Proceedings of the 2nd ACM workshop on Online social networks* (New York, NY, USA, 2009), WOSN '09, ACM, pp. 31–36.

[46] Tewari, S., and Kleinrock, L. Analytical model for bittorrent-based live video streaming. In *Consumer Communications and Networking Conference, 2007. CCNC 2007. 4th IEEE* (jan. 2007), pp. 976 –980.

[47] Tian, Y., Wu, D., and Ng, K. Modeling, analysis and improvement for bittorrent-like file sharing networks. In *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings* (2007), IEEE, pp. 1–11.

[48] Urvoy-Keller, G., and Michiardi, P. Impact of inner parameters and overlay structure on the performance of bittorrent. In *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings* (2007), IEEE, pp. 1–6.

[49] uTorrent Bittorrent Client's Forum. Available on `http://forum.utorrent.com/viewtopic.php?id=47523`.

[50] uTorrent Bittorrent Client's Good Setting. Available on `http://forum.utorrent.com/viewtopic.php?id=58404`.

[51] Valancius, V., Laoutaris, N., Massoulié, L., Diot, C., and Rodriguez, P. Greening the internet with nano data centers. In *Proceedings of the 5th international conference on Emerging networking experiments and technologies* (2009), ACM, pp. 37–48.

[52] Vu, L., Gupta, I., Nahrstedt, K., and Liang, J. Understanding overlay characteristics of a large-scale peer-to-peer iptv system. *ACM Trans. Multimedia Comput. Commun. Appl. 6*, 4 (Nov. 2010), 31:1–31:24.

[53] Vuong, Q. Likelihood ratio tests for model selection and non-nested hypotheses. *Econometrica: Journal of the Econometric Society 57*, 2 (1989), 307–333.

[54] Wang, J., Huang, C., and Li, J. On isp-friendly rate allocation for peer-assisted vod. In *Proceedings of the 16th ACM international conference on Multimedia* (New York, NY, USA, 2008), MM '08, ACM, pp. 279–288.

[55] Watts, D., and Strogatz, S. Collective dynamics of small-world networks. *Nature 393*, 6684 (1998), 440–442.

[56] Wu, D., Dhungel, P., Hei, X., Zhang, C., and Ross, K. Understanding peer exchange in bittorrent systems. In *Peer-to-Peer Computing (P2P), 2010 IEEE Tenth International Conference on* (2010), IEEE, pp. 1–8.

[57] Xu, D., Kulkarni, S. S., Rosenberg, C., and Chai, H.-K. Analysis of a cdn–p2p hybrid architecture for cost-effective streaming media distribution. *Multimedia Systems 11*, 4 (2006), 383–399.

[58] Yin, H., Liu, X., Zhan, T., Sekar, V., Qiu, F., Lin, C., Zhang, H., and Li, B. Livesky: Enhancing cdn with p2p. *ACM Trans. Multimedia Comput. Commun. Appl. 6*, 3 (Aug. 2010), 16:1–16:19.

[59] Zhang, C., Dhungel, P., Liu, Z., and Ross, K. Bittorrent darknets. In *INFOCOM, 2010 Proceedings IEEE* (2010), IEEE, pp. 1–9.