Switching Algorithms in Peer to Peer Networks - Review

M. Padmavathi[1] and R.M. Suresh[2]

[1] R.M.D. Engineering College, Kavaraipetai
[2] R.M.K. Engineering College, Kavaraipetai
mpadmavathi1979@gmail.com, rmsuresh@hotmail.com

Abstract. The peer-to-peer (P2P) network is heavily used for content distribution and is popularly used for internet file sharing. In a peer to peer network the average time to download a given file becomes a key metric for performance measure. The common approach of analyzing the average download time based on average service capacity is fundamentally flawed, and shown that the spatial heterogeneity of service capacities of different source peers and temporal fluctuations in service capacity of a single source peer have significant impact on increasing the average download time in P2P networks. This paper gives a comparative study of various downloading algorithms to effectively remove these negative factors thereby reducing the average download duration of a file.

Keywords: P2P networks, average download time, average service capacity.

1 Introduction

PEER-TO-PEER (P2P) file-sharing applications are becoming increasingly popular for content distribution and it accounts for more than 80% of the internet's bandwidth usage. The early model for content distribution was a centralized one, in which users have to compete for limited resources in terms of bottleneck bandwidth or processing power of a single server which resulted in poor performance to all users. From a single user's perspective, the duration of the download time for that individual user is the most often used performance metric.

P2P technology tries to solve the issue of scalability by making the system distributed. The aggregated output or the service capacity of a peer to peer network generally increases as the number of peers involved in uploading the same file increases in the network [1,2]. With this increasing service capacity, theoretical studies have shown that users of a P2P network enjoy much faster downloads [1][2]. The performance measurement study shows that, the per-user performance in a P2P network may be even worse than that of centralized network architecture as shown in [3]. In reality, even downloading files of less than 10 MB in size may take from 5 minutes to several hours and bigger files with size greater than 100 Mb taking hours to weeks. Further, even when all users try to download the same file, each of them may have different downloading times, depending on the available capacity

N. Meghanathan et al. (Eds.): CCSIT 2012, Part II, LNICST 85, pp. 156–166, 2012.

fluctuation, the path it chooses to download the file, etc. Thus in an ideal scenario, the downloading peer for a given bandwidth will be constrained by the limitation of the access link. In practical scenarios however there are other factors (Peer selection algorithms, free riders to name a few) which prevent a peer from fully utilizing the available bandwidth. Literature study has shown there is still scope for improvement in P2P systems. Using the average service capacity to find the performance of the network has been extensively used in literature [1] [2].

1.1 Restrictions of Average Service Capacity

Consider a P2P network, a peer downloading a file of size F from N possible source peers and C_i the average end-to-end available capacity between the downloading peer and the i^{th} source peer (i =1,2,…N). The actual value of C_i is unknown till the downloading peer actually connects to the source peer i. The average service capacity of the network, $\vec{C}$, is given by $\sum_{i=1}^{N} Ci / N$ .The average download time T , for a file of size F would be

$$T = F / \vec{C} \tag{1}$$

In practical scenarios it is found that the actual average download time is far different from the computed average download time shown in equation (1) for each user in the network.

1.2 Impact of Fluctuations in Service Capacity

Fluctuations in service capacity can be attributed to

- The spatial heterogeneity in the available service capacities of different end-to-end paths and
- The temporal correlations in the service capacity of a given source peer.

This paper considers the impact of heterogeneity. Suppose that there are two source peers with service capacities of C_1 =200 kbps and C_2 =50 kbps, respectively, and there is only one downloading peer in the network. Because the downloading peer does not know the service capacity of each source peer prior to its connection, the best choice that the downloading peer can make to minimize the risk is to choose the source peers with equal probability. For this scenario, the average capacity that the downloading peer expects from the network would be (200+50) / 2= 125 kbps. If the file size F is 1 MB, the predicted average download time is 64 seconds from (1). However, the actual average download time is 1/2(1 MB/200 kbps) + 1/2(1 MB/50 Mbps) = 66.7 seconds. It is seen that spatial heterogeneity has an impact in the average download time and actually increases the average download time. If the average service capacity is known before the downloading peer makes the connection, an obvious solution to the problem of minimizing the average download time is to find the peer with the maximum average capacity, i.e., to choose peer j with the

average capacity Cj as the average download time Ti over source peer i would be given by F / C_i. Consider again the previous two-source peer example with C_1 =100 kbps and C_2=150 kbps, As the goal is to minimize the download time, an obvious choice would be to choose source peer 2 as its average capacity is higher. Now, let us assume that the service capacity of source peer 2 is not a constant, but is given by a stochastic process C_2 (t) taking values 50 or 250 kbps with equal probability, thus giving $E\{C_2(t)\} = C_2 = 150$ kbps. If the process $C_2(t)$ is strongly correlated over time such that the service capacity for a file is likely to be the same throughout the session duration, it takes on average (1 MB/50 kbps ÷ 1 MB/250 kbps)/2 = 96 seconds, while it takes only 80 seconds to download the file from source peer 1. Thus it is evident that the average download time is increased due to heterogeneity and temporal correlations of service capacity.

The rest of the paper is organized as follows. Section 2, discusses about factors of average download time. Section 3, characterizes average download time based on these factors namely spatial heterogeneity and temporal correlations. Section 4 discusses about various switching algorithms that are in existence to remove the two negative factors. Section 5 concludes the limitations of each algorithm.

2 Factors of Average Download Time

This section describes briefly the characteristics of the service capacity that a single user receives from the network from the user's perspective. Specifically, the system considers the heterogeneity of service capacities over different network paths and the stochastic fluctuation of the capacity over time for a given source peer.

2.1 Heterogeneity of Service Capacity

In a P2P network, the service capacities from different source peers are different. There are many reasons for this heterogeneity. On each peer side, physical connection speeds at different peers vary over a wide range [11] . On the network side, peers are geographically located over a large area and each logical connection consists of multiple hops.

2.2 Correlations in Service Capacity

While the long-term average of the service capacity is mainly governed by topological parameters, the actual service capacity during a typical session is never constant, but always fluctuates over time [12],[13]. There are many factors causing this fluctuation. First, the number of connection a source peer allows is changing over time, which creates a fluctuation in the service capacity for each user. Second, some user applications running on a source peer such as online games may throttle the central processing unit and impact the amount of capacity it can offer. Third, temporary congestion at any link in the network can also reduce the service capacity of all users utilizing that link. The short-term variations in the capacity are mainly due to the window size fluctuation in TCP, while the long-term variations are due to

network congestion, changes in workload or the number of connecting users at the source peer, etc. The long-term fluctuation typically lasts over a longer time scale, say, few minutes up to several hours.

3 Characterizing Average Download Time

The formal definition of the download time is given as

Definition 1: Let C(t) denote the time-varying service capacity of a given link (path) at time $t(t = 1‹2‹▷▷▷)$ over the duration of a download. Following the definition in [14], the download time T for a file of size F is defined as

$$T = \min \{ s > 0 \mid \sum_{t=1}^{S} C(t) \geq F \} \tag{2}$$

The random variable T is the first hitting time of the process C(t) to reach level F. If $¶C(t)‹t \in \mathbb{N}◇$ are independent and identically distributed $(i▷i▷d▷)$, then by assuming an equality in (2), Wald's equation [10] gives that

$$F = E \{ \sum_{t=1}^{T} C(t) \} = E\{C(t)\}E\{T\} \tag{3}$$

The expected download time, measured in slots, then becomes $E\{T\} = F/\mathbb{E}¶C(t)◇$. Note that (3) also holds C (t) is constant (over t). It can be noted that whenever the service capacity is i.i.d. over time or constant, a direct relationship can be established between the average service capacity and the average download time.

3.1 Impact of Heterogeneity in Service Capacity

The impact of heterogeneous service capacities of different source peers is considered in this section. To decouple the effect of correlations from that of heterogeneity Wald's equation holds true for *each source peer* and this allows the average capacities to be different for various source peers.

Let N be the number of source peers in the network and Ci(t) be the service capacity of source peer at time slot . The paper assumes that Ci (t) is either constant or i.i.d. over t such that (3) holds. Let $Ci = E \{ C_i(t) \}$ be the average capacity of source peer . Then, the average service capacity the network offers to a user becomes

$$A(\vec{C}) = \frac{1}{N} \sum_{i=1}^{N} Ci \tag{4}$$

where $\vec{C} = C1, C_2, \ldots C_N$ is the arithmetic mean $A(\vec{C})$ of the sequence . Thus, one may expect that the average download time, E{T} of a file of size F would be

$$E\{T\} = \frac{F}{A(\vec{C})} \tag{5}$$

Since the user can choose one of N source peers with equal probability, the actual average download time in this case becomes

$$E\{T\} = \frac{1}{N}\sum_{i=1}^{N}\frac{F}{Ci} = \frac{F}{H(\vec{C})} \tag{6}$$

Where $H(\vec{C})$ is harmonic mean of $C_1, C_2, .. C_N$ defined by

$H(\vec{C}) = [\frac{1}{N}\sum_{i=1}^{N}\frac{1}{Ci}]$. $A(\vec{C}) \geq H(\vec{C})$, because , it follows that (6) $\geq$ (5). It is seen that the actual average download time in a heterogeneous network is always greater than that given by "the average capacity of the network" as in (5).

3.2 Impact of Correlations in Service Capacity

Consider a fixed network path between a downloading peer and its corresponding source peer for a file of size F. Let C(t) be a stationary random process denoting the available capacity over that source at time t and assumes that C(t) is positively correlated over time. Then, as before, the download time of a file (or the first hitting time of the process C(t) to reach a level F) can be defined as T_{cor}, where the subscript "cor" is a correlated stochastic process, if the system is able to remove the correlations from C(t). Let C'(t) be the resulting process and T_{ind} be the stopping time for the process to reach level F , where the subscript "ind" now means that is independent over time. Then, again from Wald's equation,

$$E\{T_{ind}\} = \frac{F}{E\{C'(t)\}} = \frac{F}{E\{C(t)\}} \tag{7}$$

If C(t) is 100% correlated over time then the download time T_{cor} becomes $T_{cor} = F / C$. Hence, Jensen's inequality shows that

$$E\{T_{cor}\} = FE\{\frac{1}{C}\} \quad \frac{F}{E\{C\}} = E\{T_{ind}\} \tag{8}$$

i.e., the average first hitting time of an 100% correlated process is always larger than that of an i.i.d. counterpart.

Theorem 1: Suppose that C(t) , $t \geq 1$ is associated. Then, $E(T_{cor}) \geq E(T_{ind})$.

Theorem 1 states that the average download time of a file from a source peer with correlated service capacity is always larger than that of an i.i.d. counterpart. From previous discussions, in general the average download time, E(T) , should be calculated using E (F/C(t)) rather than the commonly used F / E [C(t)]. The relationship between the average download time and the degree of correlation in the available bandwidth is given by a stationary 1st autoregressive process (AR1) as

$$C(t+1) = . C(t) + \varepsilon(t) + \alpha \quad (9)$$

Equation (9) shows that, the average download time increases as the degree of correlation increases, for different ρ and Є(t) .

4 Methods to Minimize File Download Duration

Intuitively, if a downloader relies on a single source peer for its entire download it results in a long download in case of slow source peer. Since the service capacity of each source peer is different and fluctuates over time, utilizing different source peers either simultaneously or sequentially within one download session would be a good idea to diversify the risk. This section briefly describes all the methods that are used to download a file form P2P networks. The methods to download a file are (i) Byte based algorithms; (ii) Time based algorithms. (iii) Choke based algorithm.

The proposed analysis assumes the following properties [4]:

i. The service capacity of a source is constant within one time slot.
ii. Each downloader selects its source independently.
iii. Each downloader makes a blind choice, i.e., the sources are randomly chosen uniformly over all available sources.

4.1 Byte-Based Algorithms

Peer-to-peer research has predominately focused on byte based switching algorithms to alleviate the amount of time clients remain with underperforming nodes [1][2][3].

Parallel Downloading

Parallel downloading is one of the most noticeable way to reduce the download time [5],[7]. In parallel download, a file of size F is divided into k chunks of equal size and the single file is allowed to download in parallel with k simultaneous connections yielding a download duration of $\max\{t_1, t_2, \ldots, t_k\}$ (where t_k represents the time chunk k took to download during time t). If parallel downloading is used to download a file of size F from the network with only two source peers of service capacities C_1,C_2 respectively, the download time is given by

$$T_P = \max\left\{\frac{F}{2C1}, \frac{F}{2C2}\right\} = \frac{F}{2C1} \quad (10)$$

From (1) the predicted download time is given by

$$E\{T\} = \frac{F}{A(\vec{C})} = \frac{2F}{C1+C2} \quad (11)$$

Thus, even in the network with one user, parallel downloading may not reduce the download time as the performance of parallel download depends upon the distribution of the underlying service capacities and could be much worse than the ideal case. By

making the chunk-size proportional to the service capacity of each source peer, parallel downloading can yield the optimal download time. But such scheme requires global information of the network.

Limitations : This method exposes the entire download session to the slowest server.

Random Chunk-Based Switching

In the random chunk-based switching scheme[4], the file is divided into many small chunks and a user downloads chunks sequentially one at a time. When a peer completes downloading a chunk from the current source peer, the next source peer is selected randomly to download the next chunk. In this way, if the downloader is currently stuck with a bad source peer, it will stay there for only the amount of time required for finishing one chunk. The download time for one chunk is independent of that of the previous chunk. Intuitively, switching source peers based on chunk can reduce the correlation in service capacity between chunks and hence reduce the average download time. Suppose if there is no temporal correlation in service capacity and a file of size F is divided into m chunks of equal size, and let T_j be the download time for chunk . Then, the total download time is given by $T_{Chunk} = \sum_{j=1}^{m} t_j$ .Since each chunk randomly chooses one of source peers the expected download time will be given by

$$E\{ T_{Chunk} \} = \sum_{j=1}^{m} \frac{1}{N} \sum_{i=1}^{m} \frac{F/m}{C_i} = \frac{F}{H(\vec{C})} \quad (12)$$

The result in (10) is identical to the download time given in (6) where a user downloads the entire file from an initially randomly chosen source peer.

Limitations: In the chunk-based switching, there is possibilities of getting stuck with a source peer of very low service capacity, so that downloading a fixed amount of bytes from the source peer may still take a long time. This long wait can be avoided by making the size of each chunk very small, but this would cause too much overhead associated with switching to many source peers and integrating those many chunks into a single file. This shows that the chunk-based switching is still subject to the curse of bad spatial heterogeneity .

Random Permanent Connection

In permanent connection, the source selection function does not change in time t . When the list of available source peers is given, the downloader will choose one of them randomly with equal probability and will stay with that source peer *permanently [4]* until the download completes.

Limitations: Since the downloader randomly chooses a source peer at each time slot, it is possible for the downloaded to get stuck with bad source peer for that fixed time slot t.

4.2 Time- Based Algorithms

Time-based algorithms allow clients to select a new server based on expiration of a timer rather than using chunks, parallel, or permanent algorithms. Byte-based algorithms rely on average service capacity as a basis for calculating download duration. But temporal fluctuations in network capacity as well as inconsistencies in using average capacity provide for a new model. To accurately reproduce temporal fluctuations, a stochastic model can be used.

Random Periodic Switching

In random periodic switching [9], the downloader randomly chooses a source peer at each time slot t, independently of everything else. In other words, the source selection function U(t) forms an i.i.d. sequence of random variables, each of which is again uniformly distributed over (1,2,…N). Fig. 2 illustrates the operation of the source selection function U(t) for random periodic switching. In this figure, source 1 is selected at time 1, source N is selected at time 2, and so on.

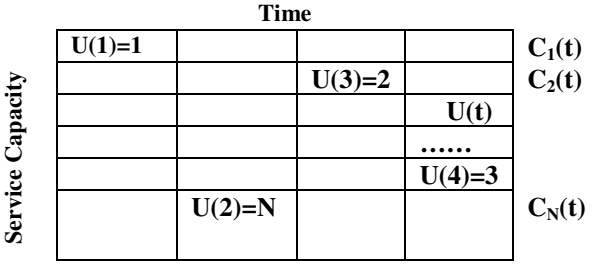

Fig. 1. Source Selection function in random periodic switching

By using random periodic switching, it is always possible to make the capacity process *very lightly correlated*, or almost independent. Fig. 2, shows that the average download time for a lightly correlated process is very close to that given by Wald's equation.(i.e) the time-based switching shows a more accurate representation of download duration in both single client and multi client scenarios.

Limitations: Time-based switching does not address temporal fluctuations during the allotted download time period [10].The initial problem of clients remaining with underperforming servers was not fully addressed in [4].

4.3 Choke-Based Algorithms

Choke based algorithms incorporates a preemptive choke at the client level that enables a client to independently identify a choke threshold that can be used for intelligent departure from poor performing servers. Additionally, the choked-based algorithm increases the accuracy of prediction of the file download duration[10].

Choke Based Switching

Choking algorithm is initialized by performing a global catalog lookup. This initial seeding of the choking algorithm will provide average overlay capacity and is set to 200 Kbps. The chokepoint for current capacity is determined by the choking algorithm (9) with the seeded value and any previous download session capacities. For example, let N be the number of contacted node-servers while C_i is the bandwidth between the client and the node-server for i= 1, 2…N gives

$$CP = \frac{1}{N} \sum_{i=0}^{N} C_i \tag{13}$$

At the start of every connection, the client will calculate a new chokepoint value using (9). The chokepoint will be evaluated against a margin (θ) to determine if the server has enough capacity where θ = {0.05, 0.1 … 1}. For example, if the chokepoint is 80Kbps and the server capacity is 50Kbps with a θ = .5, the chokepoint threshold is (80Kbps – (80Kbps*.5)) = 40Kbps. So the server would be accepted [8]. For every degree of heterogeneity (δ), all margin variables θ = {0.2, 0.4 … 1.0} are simulated. As δ approaches 1, the download duration reduces regardless of θ. The severity of the decrease is controlled by θ. For example when θ = 0.2, there is a decrease in download duration from 95 minutes to 58 minutes. Over the same span, time based switching remains fairly stable at 100 minutes. Experiments involving multiple clients/multiple servers investigation shows that choke based algorithms minimizes the file download duration rapidly compared to time based algorithms.

Table 1. Multi Client results for various switching Algorithms

Clients	Byte Based(min)	Time based (min)	Choke based (min)
100	25440	106	100
200	44160	184	182
300	65760	274	270
400	87360	364	360
500	110880	462	448
600	133920	558	541

The table 1 indicates that as the number of clients increases, the ability of the choke-based algorithm to cope with the congestion increases. The choke threshold margin, θ, is also significant. As the θ increases to 1 (no choke), there is the expected increase in download duration. During the simulations, the smallest values of θ (less that .15 margin) rarely produced the fastest download duration.

5 Conclusions

This paper surveys on minimizing the average download duration of each user in a P2P network. With the devastating usage of network resources by P2P applications in the current Internet, it is highly desirable to improve the network efficiency by

reducing each user's download time. In contrast to the commonly-held practice focusing on the notion of average capacity, the spatial heterogeneity and the temporal correlations in the service capacity can significantly increase the average download time of the users in the network, even when the average capacity of the network remains the same. This paper investigated several "byte-based" (file size based) schemes widely researched and used in practice, including chunk-based file transfer, parallel downloading and permanent connections. Investigations reveal that all byte-based schemes are not so effective in reducing the two negative factors that increase the average download time. Time-based switching improves upon byte-based algorithms by alleviating the reliance on average capacity of the overlay for accurate download duration estimation using a stochastic AR model. The investigation revealed time-based switching does not address temporal fluctuations during the allotted download time period. Another major area of concern in time based algorithms is the initial problem of clients remaining with underperforming servers. Choke based algorithm enhances time based algorithms by incorporating a preemptive choke at the client level that enables a client to independently identify a choke threshold that can be used for intelligent departure from poor performing servers. It is shown that in both single client and multi client scenarios that the choke-based algorithm decreases clients download duration. Additionally, the choked-based algorithm increases the accuracy of prediction of the file download duration. Future work will include investigation into client modification of the threshold margin (θ) based upon current network conditions.

References

[1] Yang, X., de Veciana, G.: Service capacity of peer to peer networks. In: Proc. IEEE INFOCOM, pp. 2242–2252 (March 2004)
[2] Qiu, D., Srikant, R.: Modelling and performance analysis of Bittorrent-like peer-to-peer networks. In: Proc. ACM SIGCOMM (August 2004)
[3] Gummadi, K.P., Dunn, R.J., Saroiu, S.: Measurement, modeling, and analysis of a peer-to-peer file sharing workload. In: ACM Symp. Operating Systems Principles, SOSP (2003)
[4] Chiu, Y., Eun, D.: Minimizing file download time in peer to peer networks. IEEE/ACM Transactions on Networking 16(2), 253–266 (2008)
[5] Ng, T., Chu, Y., Rao, S., Sripanidkulchai, K., Zhang, H.: Measurement- based optimization techniques for bandwidth-demanding peer-to-peer systems. In: Proc. IEEE INFOCOM, pp. 2199–2209 (April 2003)
[6] Koo, S.G.M., Rosenberg, C., Xu, D.: Analysis of parallel downloading for large file distribution. In: Proc. IEEE Int. Workshop on Future Trends in Distributed Computing Systems (FTDCS), pp. 128–135 (May 2003)
[7] Ramachandran, K.K., Sikdar, B.: An analytic framework for modeling peer to peer networks. In: Proc. IEEE INFOCOM, pp. 215–269 (March 2005)
[8] Lehrfeld, M.R., Simco, G.: Choke-based Switching Algorithm in Stochastic P2P Networks to Reduce File Download Duration. In: IEEE 2010, pp. 127–130 (2010)
[9] Saroiu, S., Gummadi, K.P., Gribble, S.D.: A measurement study of peer-to-peer file sharing systems. In: ACM Multimedia Computing and Networking, MMCN (2002)

[10] Jain, M., Dovrolis, C.: End-to-end estimation of the available bandwidth variation range. In: Proc. ACM Sigmetrics (June 2005)
[11] Hu, N., Steenkiste, P.: Evaluation and characterization of available bandwidth probing techniques. IEEE J. Sel. Areas Communication 21(6), 879–894 (2003)
[12] Koo, S.G.M., Kannan, K., Lee, C.S.G.: Neighbor-selection strategy in peer-to-peer networks. In: Proceedings of IEEE International Conference on Computer Communications and Networks (ICCCN), Rosemont, IL (October 2004)
[13] Sherwood, R., Braud, R., Bhattacharjee, B.: Slurpie: A cooperative bulk data transfer protocol. In: Infocom, Hong Kong (March 2004)