EEE3034 - Mediacasting Assignment

Content Delivery Networks, Peer to Peer Networks and Collaborative Filtering

Eva Esteban

Department of Electrical and Electronic Engineering

Faculty of Engineering and Physical Sciences

University of Surrey

2018/19

Abstract

The aim of this report is to provide an in-depth analysis of the main technologies which are currently used by companies to provide content delivery to their customers, as well as to explore the Weighted Slope One method for predicting the ratings that customers are likely to give to a range of different movies. The first part of the document focuses on the data transfer capability of Content Distribution Networks (CDNs), Peer to Peer (P2P) Networks and Hybrid CDN-P2P networks used for content delivery. Additionally, the second part of this report analyses the efficiency of the Weighted Slope One method as a Collaborative Filtering approach to predict user ratings for media contents.

Contents

1 INTRODUCTION

Nowadays, different methods are employed by companies to successfully deliver their content to all the users that manifest an interest in receiving it around the world. This process is known as content delivery. Two of the main technologies that are currently used to provide reliable content delivery are Content Delivery Networks (CDNs) and Peer to Peer (P2P) Networks. These technologies can be combined into a third type of networks known as Hybrid CDN-P2P Networks.

CDNs can be defined as geographically distributed networks that consist of proxy servers and their data centers. They focus on providing both high availability and high performance to Internet users and are characterised by being arranged in a tree structure [6]. In such a structure, the content from an origin server propagates to the leaf nodes, which then deliver this content to the users that are geographically close to them. On the other hand, P2P networks rely on the ability of users to distribute the content between each other, hence reducing the load of the origin server and defining peers as equally privileged participants of the system. When combined, CDNs and P2P technologies form a Hybrid CDN-P2P network, which integrates the reliability of CDNs with the scalability of P2P networks [11]. Further information and analyses on these techniques can be found in subsequent sections of this document.

Once companies ensure that their content is delivered successfully in a robust and secure way, they use different techniques to increase the users' consumption of this content, which, in turn, increases the companies' sales and profits. One of these techniques consists of recommending products to the users that they are likely to consume. There are three main categories of recommendations systems that are nowadays used for this: Collaborative Filtering (CF), Content-Based and Hybrid methods. The following report will focus on the Collaborative Filtering methods, which allow large companies such as Netflix [10] to predict the rating of unbought products that a client might be interested in. This prediction is based on the ratings given by other clients with similar preferences to the target client.

2 Content Delivery Networks (CDNs) and Peer to Peer (P2P) Networks

Content Delivery Networks (CDNs) and Peer to Peer (P2P) networks are two of the existing technologies used by companies to deliver their content efficiently to users. In order to evaluate the performance of both techniques independently and combined into a Hybrid CDN-P2P network, the programs CDN_only P2P_only, and CDNP2P were run with the parameters set in the configuration files. The parameter values specified in the configuration files were: CDN_Nodes (the number of CDN servers), CDN_Upload_Bandwidth (the upload bandwidth of one edge server measured in bits/cycle), Peer_Nodes (the number of client or peer nodes), Peer_Node_Upload_Bandwidth (the upload bandwidth of one edge server measured in bits/cycle), Peer_Node_Download_Bandwidth (the download bandwidth of one edge server measured in bits/cycle), File_Size (an integer representing the number of bits in the file), and Number_of_file_segments (the number of segments in which a file is partitioned). By varying these parameters, an analysis was performed on how varying the number of edge servers and their upload bandwidth affects the waiting time in a CDN-only network, how increasing the number of users and changing the number of file segments affects the waiting time in a P2P-only network, the effect that modifying the number of edge servers and increasing the number of users has on a Hybrid CDN-P2P network. The first part of the content delivery section of this report focuses on a high-level analysis of the waiting time obtained for the different parameters, while the second part provides a more in-depth analysis of the waiting time taken for different numbers of users to obtain the files. For both of these sections, comparisons with the theoretical predictions are provided.

A Content Delivery Network (CDN) is a large system of servers that are distributed in different areas of a country or around the world. This system of servers delivers online pages and other Web content to the users interested in them based on their geographic location. This ensures that clients will download the files that are near them, which in turn reduces latency and routing problems. An example of a CDN is the Akamai CDN [1]. It consists of approximately 100,000 servers located in over 1,000 networks in 70 different countries. These serve approximately 20% of the web traffic and power iTunes [2], BBC iPlayer [3] and other systems [11].

A Peer to Peer (P2P) Network provides an architecture in which the workload is partitioned between peers. This allows the network to exploit the ability of the users to distribute content, hence increasing the network's uploading capability. In P2P networks, when a user obtains a file or part of a file, they can send it to another user. This, in turn, increases the uploading bandwidth of the P2P system. Thus, the users that send content to other users become both a client and a server, as opposed to the traditional client-only architecture of the users provided by CDNs.

Additionally, it is possible to have a Hybrid CDN-P2P network which incorporates both the reliability of CDN networks and the additional scalability of P2P networks, applied by segmented files during busy file transferring times [11]. These provide an efficient approach to implement processes such as large-scale video distribution [13].

2.1 Methodology and Simulation Parameters

The data was gathered during the laboratory session using the Linux Operating System [7] to run the CDN-only, P2P-only and P2P-CDN programs. It was analysed using Matlab software [8] to optimise the process of extracting the necessary parameters and performing the required calculations. Matlab was also employed to plot the data on different graphs in order to optimise the analytics process. Additionally, the results were stored using Microsoft Excel software [9], as well as the University of Surrey GitLab platform [14]. The data used for this section can be found on the GitLab repository `https://gitlab.eps.surrey.ac.uk/ee00157/mediacasting_cwk`.

2.1.1 CDN-ONLY NETWORKS

The process of fetching data from a CDN consists of a host downloading the data from the CDN node that is geographically closest to it. This process is performed by using Domain Name System (DNS) redirection, procedure in which a CDN name server sends different IP addresses for the CDN nodes depending on where the client requesting the data is located [11].

However, CDNs have certain limitations. In CDNs the bandwidth of the edge servers constrains the system's total bandwidth available for the delivery of content. If we want to serve a file to N users or peers which consists of S number of bits by using E number of edge servers with a peer node upload bandwidth of U bits/sec, the total upload bandwidth of the CDN system will be given by E*U bit/sec [11]. If each of the N users has a download bandwidth of D bits/sec, the CDN system will only be able to serve the files to a maximum number of $\frac{EU}{D}$ users at the same time. It will take S/D seconds for each of this users being served to obtain the complete file. Thus, the time taken to serve all the N users will be given by $f(N) = \frac{ND}{EU} * \frac{S}{D}$ which can be simplified to $f(N) = \frac{NS}{EU}$ [11]. From the expression it can be noted that the waiting time f(N) for a number of users to obtain a file can be reduced by increasing either the upload bandwidth of the CDN system U or the number of edge servers E.

For the CDN-only section of this work, both the upload bandwidth of the system U and the number of edge servers E were modified. Given that this analysis is concerned with analysing the CDN itself and its properties and since the number of users N is a factor external to the network, N was set to a maximum value of 200. This allowed an objective evaluation of the network's behaviour for the different E and U values, which are internal properties of the CDN itself. The same criteria applies to the download bandwidth of the peers D, which was set to a constant value of 10 bits/sec. The different values tested for the number of edge servers were 1, 10, 50, 100, 500 and 1,000. These values were all tested for a CDN upload bandwidth of 10 bits/cycle. Additionally, for each of the six fixed numbers of edge servers, the values of 10, 50, 100, 500 and 1,000 bits/cycle were tested as the upload bandwidth of the system, which is consistent with the bandwidths usually presented by systems in practice.

2.1.2 P2P-ONLY NETWORKS

For a P2P system with $N = 2^k$ users who all have an upload and download bandwidth of D bits/sec, if a user has a file consisting of S bits to serve to the other users, we can calculate the time it will take for all the users to obtain the complete file. Given that we need $T = \frac{S}{D}$ seconds to download the file, after the first T seconds 2 users will have the file. After 2T seconds, 4 users will have the file. After 3T seconds, 8 users will have the file. Thus, after kT seconds, 2^k users will have the file. Since $N = 2^k$ is the number of users, $k = log_2(N)$. From this we can deduce that the waiting time is given by $f(N) = \frac{S}{D}log_2(N)$ [11]. Therefore, as the number of users changes, the waiting time f(N) is expected to vary with a logarithmic complexity, which provides advantages against the linear complexity that characterises CDNs. In order to analyse the variation of the waiting time, for the first part of this task the CDN server was set to 1, the number of file segments was set to 1, both the download and upload bandwidths of the peers were set to 10 bits/cycle, and the number of peer nodes N was set to the following different values: 1,000, 2,000, 3,000, 4,000, 5,000, 6,000, 7,000, 8,000, 9,000 and 10,000 to simulate a real network.

In order to optimise the delivery process, a file can be split into different equal segments. Given that the number of segments if F, for an S-bit file each segment will have a size of S/F. For a system with 2^k users with an upload bandwidth of 10 bits/sec and a download bandwidth of 10 bits/sec, the user (source) with the complete S-bit file to share will send the first segment of the file to another user, the second segment of the file to a different user, the third segment of the file to a third user, and the rest of the segments in the same way until all of them have been forwarded. After k steps, 2^k users will have the first segment, at the next step 2^k users will have the second segment and so on. For N users $log_2 N$ interactions are required for everyone to receive the first segment, after which there are F-1 segments left. The total time to deliver the file will thus be given by $f(N) = \frac{S}{DF}(log_2(N) + F - 1)$, where S is the bit size of the complete file, D is the peer node download rate measured in bits/sec or bits/cycle, N is the number of users and F is the number of segments [11]. For the second part of the P2P network analysis, the values for the server node, the upload bandwidth and the download bandwidths were maintained as above. The number of file segments was varied to 1, 2, 3, 4, 5, 6, 7, 8, 9 and 10, which constitute a realistic and broad range of samples.

2.1.3 CDN-P2P NETWORKS

Hybrid CDN-P2P networks are implemented by companies in order to incorporate the implement a reliable CDN-based network whose bandwidth can be extended via P2P technology. This is useful when delivering high-demand video content to a large number of users. Hybrid CDN-P2P networks behave like CDN networks under normal conditions, and apply P2P techniques during peak times. For the first part of the analysis of Hybrid networks in this work, the number of file segments was fixed to 1 and the number of clients was set to 2,000. Under these conditions, the number of CDN nodes was tested at the configurations 10, 50, 100, 500 and 1,000.

For the second part of this analysis, the number of peers was maintained as above and the number of CDN edge servers was set to 10. The number of file segments was varied between 1, 2, 3, 4, 5, 6, 7, 8, 9 and 10. These and the above-mentioned parameters were chosen in analogy to the values tested for the CDN-only and P2P-only networks to provide an objective comparison between them.

2.2 LABORATORY DATA DISCUSSION

2.2.1 CDN-ONLY NETWORKS

Regarding the CDN-only network, for the fixed values of 200 peer nodes or users, 10 bits/cycle CDN upload bandwidth, 10 bits/cycle peer node download bandwidth, 1,000 bits file size and 1 file segment, the waiting time was measured for different numbers of CDN or edge servers. The results can be found on Table 1. These results were compared with the theoretical waiting time for different number of edge servers calculated by using the mathematical expression previously presented in section 2.1.1 of this document $f(N) = \frac{ND}{EU} * \frac{S}{D}$ simplified to $f(N) = \frac{NS}{EU}$, which can be found on Figure 1.

Number of Edge Servers	Waiting Time (time units)
1	20,001
10	2,001
50	401
100	201
500	101
1,000	101

Table 1: CDN-only network waiting time to deliver the file to all the users for different numbers of edge servers.

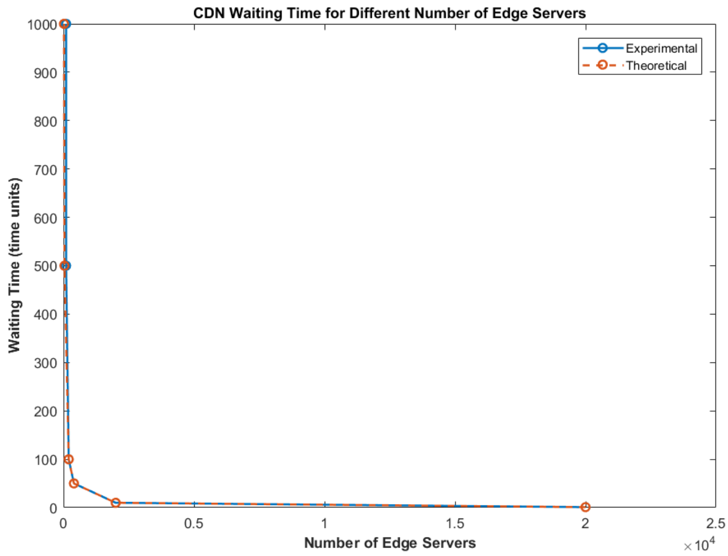

Figure 1: Graphical representation of CDN-only network waiting time to deliver the file to all the users for different numbers of edge servers.

As can be seen from Table 1 and Figure 1, as the number of edge servers increases the waiting time for delivery of the file to all the users or peers decreases. This matches the theoretical prediction obtained from the mathematical expression mentioned in section 2.1.1 of this document $f(N) = \frac{NS}{EU}$ to obtain the waiting time in a CDN. The waiting time reaches a minimum limit at approximately 500 edge servers and does not decrease any further for larger number of users in the experimental values, as opposed to the theoretical predictions where the waiting time continues decreasing. This is due the fact that the time taken for a single user to download the file from an edge server is $\frac{S}{D} = \frac{1{,}000}{10} = 100$ time units. The offset of 1 unit in the experimental value of 101 time units obtained from the 100 theoretical time units is due to the fact that the program used starts the time unit count at 1 instead of 0, and is considered negligible for this analysis. This applies to the rest of the measurements gathered during this work. Therefore, when the number of edge servers has increased enough to match the number of users, the waiting time cannot be decreased further by adding more edge servers. To reduce the waiting time in this situation, either the file size would need to be decreased or the peer node download bandwidth would need to be increased.

In order to analyse the upload bandwidth for a fixed value of 200 users together with the above-mentioned values for the other parameters, the waiting time was measured for each different number of edge servers for the five different values of the system upload bandwidth. The resulting measurements can be found on Tables 2 to 7. A graphical comparison of these results compared against the theoretical predictions obtained from the CDN equation can be found on Figure 2.

Upload Bandwidth (bits/cycle)	Waiting Time (time units)
10	20,001
50	4,001
100	2,001
500	401
1,000	201

Table 2: CDN-only network waiting time to deliver the file to all the users for 1 edge server and different values of upload bandwidth.

Upload Bandwidth (bits/cycle)	Waiting Time (time units)
10	2,001
50	401
100	201
500	101
1,000	101

Table 3: CDN-only network waiting time to deliver the file to all the users for 10 edge servers and different values of upload bandwidth.

Upload Bandwidth (bits/cycle)	Waiting Time (time units)
10	401
50	101
100	101
500	101
1,000	101

Table 4: CDN-only network waiting time to deliver the file to all the users for 50 edge servers and different values of upload bandwidth.

Upload Bandwidth (bits/cycle)	Waiting Time (time units)
10	201
50	101
100	101
500	101
1,000	101

Table 5: CDN-only network waiting time to deliver the file to all the users for 100 edge servers and different values of upload bandwidth.

Upload Bandwidth (bits/cycle)	Waiting Time (time units)
10	101
50	101
100	101
500	101
1,000	101

Table 6: CDN-only network waiting time to deliver the file to all the users for 500 edge servers and different values of upload bandwidth.

Upload Bandwidth (bits/cycle)	Waiting Time (time units)
10	101
50	101
100	101
500	101
1,000	101

Table 7: CDN-only network waiting time to deliver the file to all the users for 1,000 edge servers and different values of upload bandwidth.

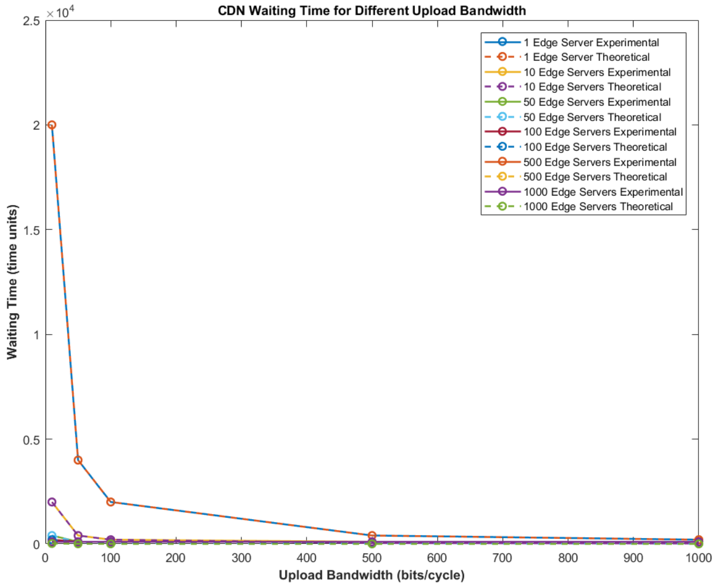

Figure 2: Graphical representation of CDN-only network waiting time to deliver the file to all the users for different numbers of edge servers and upload bandwidths.

It can be seen that the waiting time to deliver the file to all the users decreases as the upload bandwidth of the edge servers increases. This decrease of the waiting time matches the results obtained from the theoretical formula for the waiting time in CDNs. At a certain time, the waiting time stops decreasing in the experimental values remaining at 101 time units, as opposed to the theoretical waiting time values which keep decreasing. The explanation for this is analogous to the one presented in the previous section - when the number of edge servers has increased enough to match the number of users, the waiting time cannot be decreased further by adding more edge servers and the waiting time reaches a minimum limit at the time it takes a single user to download the file from an edge server. In this scenario this time is given by $\frac{S}{D} = \frac{1{,}000}{10} = 100$ time units.

2.2.2 P2P-only Networks

Regarding the P2P-only network, for the fixed values of 1 edge server, 10 bits/cycle peer node upload bandwidth, 10 bits/cycle peer node upload bandwidth, 10 bits/cycle peer download bandwidth, 1,000 bits file size and 1 file segment, the waiting time was measured for different numbers of peers. The results can be found on Table 8 and Figure 3.

Number of Peers	Waiting Time (time units)
1,000	1,001
2,000	1,101
3,000	1,201
4,000	1,201
5,000	1,301
6,000	1,301
7,000	1,301
8,000	1,301
9,000	1,401
10,000	1,401

Table 8: P2P-only network waiting time to deliver the file to all the peers for different number of peers.

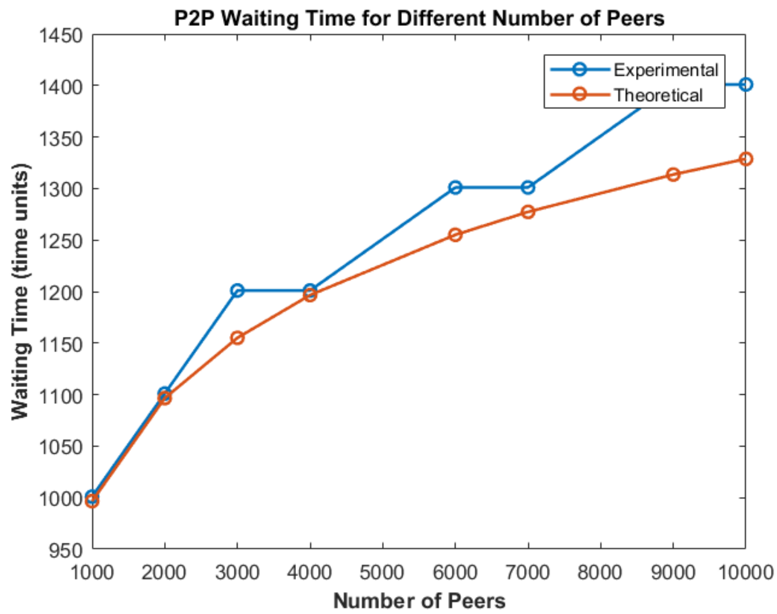

Figure 3: Graphical representation of P2P-only network waiting time to deliver the file to all the users for different numbers of peers.

An increase can be observed in the waiting time as the number of peers rises in the P2P network. This increment follows a logarithmic trend. Consequently, for small differences in small peer numbers the waiting time varies considerably, however, as the number of peers increases to large values the incremental difference in the waiting time gets smaller. This correlates well with the mathematical expression with logarithmic complexity provided in section 2.1.2 of this document $f(N) = \frac{S}{D} log_2(N)$, which describes the waiting time for N users with a download bandwidth of D to obtain a non-segmented file of size S bits.

Furthermore, for a fixed value of 2,000 peers together with the values stated above for the other parameters, the waiting time was measured for different values of the file segments. The resulting time measurements are gathered on Table 9 and graphically represented on Figure 4.

Number of File Segments	Waiting Time (time units)
1	1,101
2	851
3	749
4	651
5	601
6	579
7	556
8	534
9	529
10	481

Table 9: P2P-only network waiting time to deliver the file to all the users for different number of file segments.

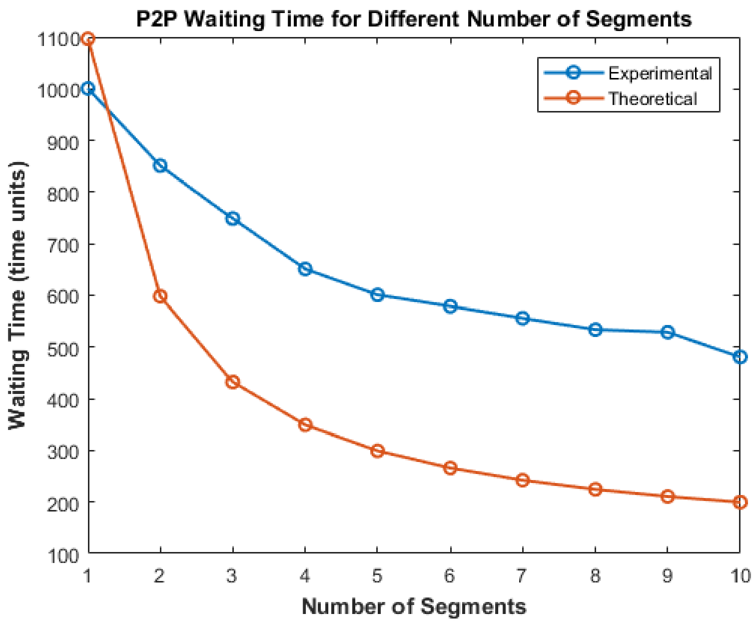

Figure 4: P2P-only network graphical representation of the waiting time to deliver the file to all the users for different number of file segments.

It can be observed that for increasing number of file segments the waiting time decreases in a near-logarithmic fashion. This trend satisfies expression for calculating the waiting time in a P2P network for different numbers of file segments, given by $f(N) = \frac{S}{DF}(log_2(N) + F - 1)$ where S represents the number of file bits, N is the number of users, D describes the download bandwidth of each user and F represents the number of segments the file is divided into. However, the experimental values do not exactly match the theoretical predictions. This inaccuracy is due to the fact that when transferring file segments in practice, the best route and timings not always chosen and as a result some nodes will have to wait for the segments for long periods of time. This is not accounted for in the theoretical model.

2.2.3 CDN-P2P Hybrid Networks

Regarding the Hybrid CDN-P2P network, for the fixed values of 2,000 peer nodes, 10 bits/cycle peer node upload bandwidth, 10 bits/cycle peer node download bandwidth, 1,000 bits file size and 1 file segment, the waiting time was measured for different numbers of edge servers. The results can be found on Figure 10.

Number of Edge Servers	Waiting Time (time units)
1	1,101
10	801
50	601
100	501
500	301
1,000	201

Table 10: Hybrid CDN-P2P network waiting time for different number of edge servers.

The data gathered shows that for a singular segment i.e non-segmented file for an increasing number of edge servers the waiting time to deliver the file to all the users decreases. Since the file is not segmented, the P2P capability of the network is not being used and therefore the network behaves like a CDN-only network adhering to the expression $f(N) = \frac{NS}{EU}$. For the particular edge server chosen in this document, the waiting time before all the peers receive the file always decreases. As an extension to this work, additional edge server values should be tested to determine if this is always the behaviour of the system.
In addition, for a fixed value of 10 edge servers and the parameters specified above, the waiting time was measured for different values of the file segments. The results of this analysis can be found on Table 11.

Number of File Segments	Waiting Time (time units)
1	801
2	601
3	545
4	476
5	441
6	426
7	406
8	391
9	385
10	351

Table 11: Hybrid CDN-P2P network waiting time for different number of file segments.

As shown on Table 11, for an increasing number of file segments with a fixed number of both peers and edge servers the waiting time of the system decreases in a logarithmic fashion. In this situation, given that the file is segmented, the P2P distributing capabilities of the network are being used. Hence, the waiting time adheres to the mathematical expression $f(N) = \frac{S}{DF}(log_2(N) + F - 1)$ explained previously.

2.3 Comparison with Theoretical Predictions

2.3.1 CDN-only Networks

Given a CDN-only network, we can calculate both the total time required to serve N users an S-bit long file and predict how many users will have obtained the file at a given time. This can be done by using the expression $f(N) = \frac{ND}{EU} * \frac{S}{D}$ simplified to $f(N) = \frac{NS}{EU}$. Here, N is the number of users to serve, S is the number of bits in the file, D is the download bandwidth of each user measured measured in bits/sec or bits/cycle, U is the upload bandwidth of each user measured in bits/sec or bits/cycle, and E is the number of edge servers.

For this analysis, the number of users that received the file after different time intervals was compared for the CDN network when varying both the edge servers and the upload bandwidth independently. Figures 5 to 10 show both the theoretical and the experimental plots representing the number of users that obtained the file against the time elapsed for edge server E values of 1, 10, 50, 100, 500 and 1,000, and for an upload bandwidth of 10, 50, 100, 500 and 1,000 bits/cycle for each number of edge servers. For these plots the file is 1,000 bits long, and the number of users N is fixed to 200.

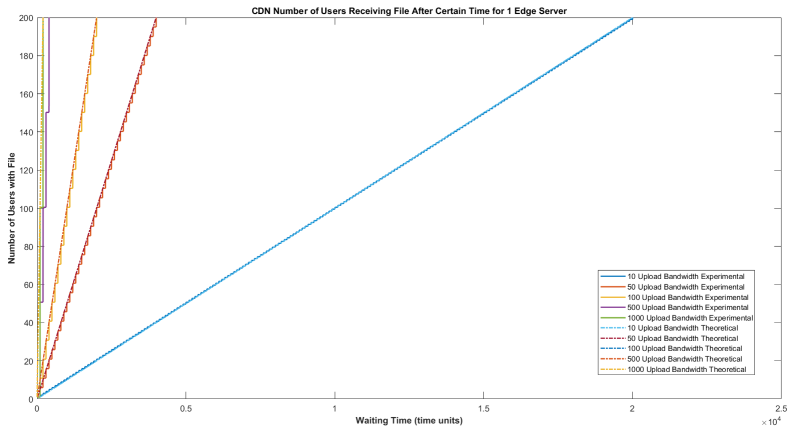

Figure 5: CDN-only network graphical representation of the number of users with the file for 1 edge server and different upload bandwidths.

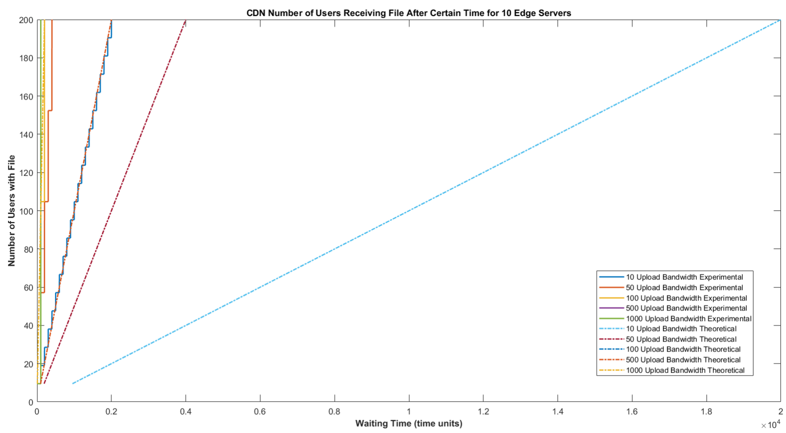

Figure 6: CDN-only network graphical representation of the number of users with the file for 10 edge servers and different upload bandwidths.

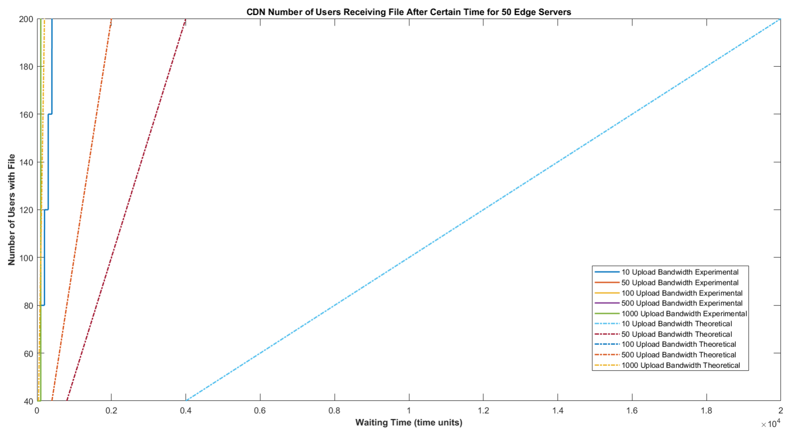

Figure 7: CDN-only network graphical representation of the number of users with the file for 50 edge servers and different upload bandwidths.

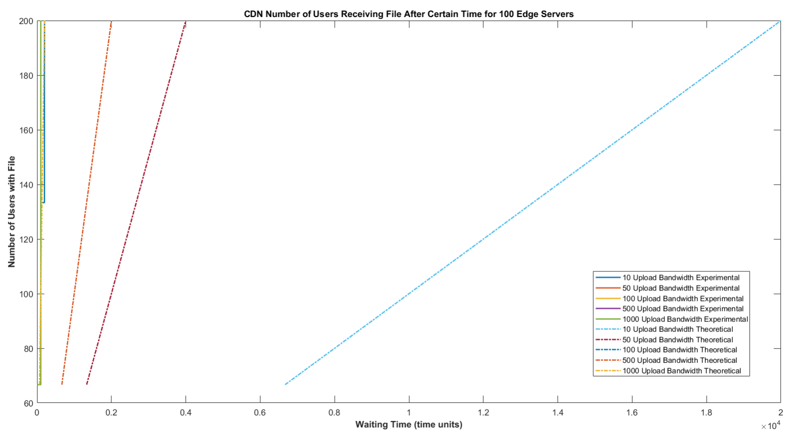

Figure 8: CDN-only network graphical representation of the number of users with the file for 100 edge servers and different upload bandwidths.

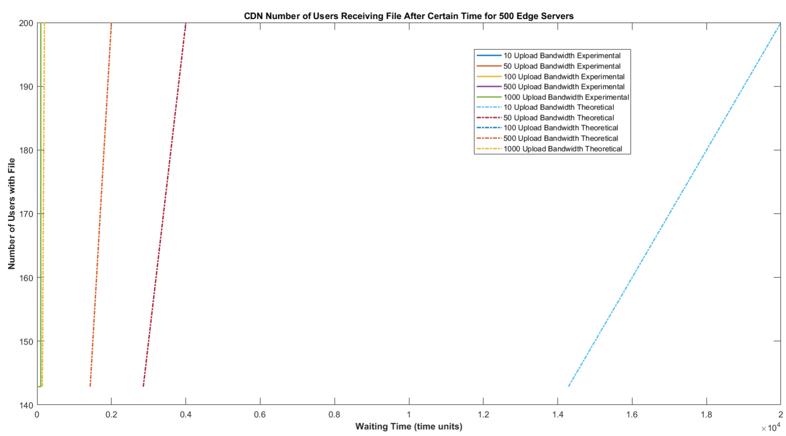

Figure 9: CDN-only network graphical representation of the number of users with the file for 500 edge servers and different upload bandwidths.

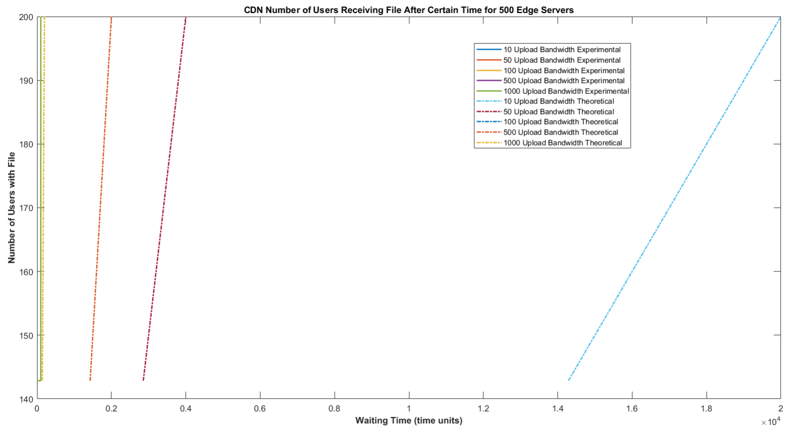

Figure 10: CDN-only network graphical representation of the number of users with the file for 1000 edge servers and different upload bandwidths.

It can be seen that for both the theoretical and the experimental values the number of users that have received the complete file increases faster when either the number of edge servers is increased, the upload bandwidth of the system is increased, or both are increased. This increase is linear, although while the theoretical values show a continuous trend, the experimental values increase in steps. This is due to the fact that it takes a fixed amount of time for a user to download a file from an edge server, given by S/D, where S if the file size in bits and D is the user download bandwidth measured in bits/second or bits/cycle. Thus, even though an increased number of nodes can serve more users, the number of users that have obtained the file will increase every 100 time units, which is $\frac{S}{D} = \frac{1000}{10} = 100$ time units approximately. Furthermore, the experimental values hit a minimum waiting time threshold at 101 time units, which causes several curves to stack on top of each other with only the top one visible in the graph. The reasoning behind this is the in-depth explanation provided in the previous section 2.2 of this report.

2.3.2 P2P-only Networks

In a P2P-only network, the time necessary to serve an S-bit file to N users each with a download bandwidth of D measured in bits/sec or bits/cycle can be calculated by using the mathematical expression: $f(N) = \frac{S}{D} log_2(N)$. This provides logarithmic complexity, which is more efficient than the linear complexity that characterises CDN-only networks.

If the file is segmented into F equal segments, with each segment therefore having a size of S/F bits, the total time to serve N users is given by: $f(N) = (\frac{S}{DF})(log_2(N) + F - 1)$, where S is the file size in bits, D is the peer node upload bandwidth measured in bits/sec or bits/cycle.

For this analysis, the number of users that received the file after different time intervals was compared for the P2P network for an increasing number of peers, as well as for a fixed number of users and increasing number of file segments from 1 to 10. The results of these analyses can be found on Figure 11.

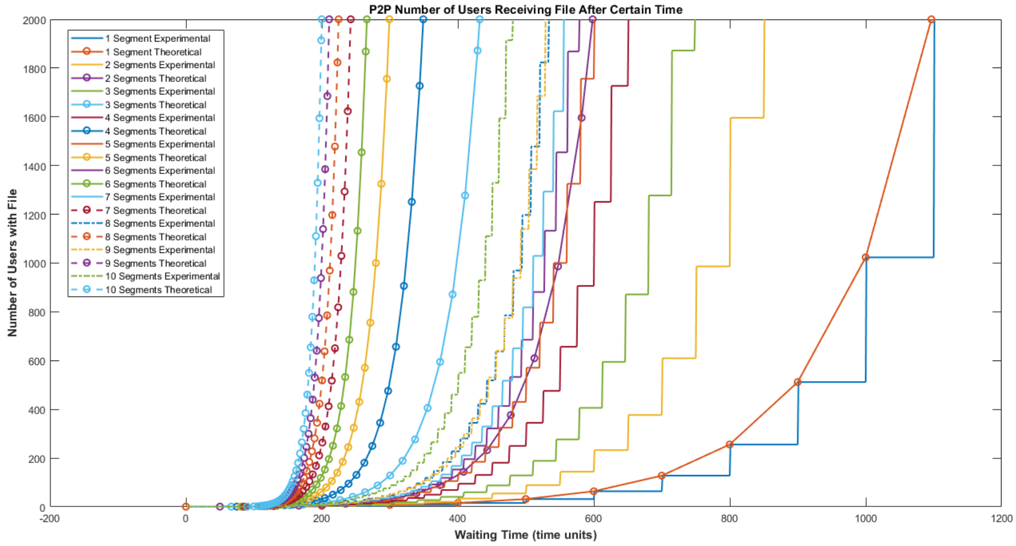

Figure 11: P2P-only network graphical representation of the number of users with the file for 1 and multiple file segments.

As can be seen from Figure 11, for the P2P network the greater the number of users the sharper the increase of the curve, which indicates that the percentage of users that have received the complete file will increase faster the more users are participating in the network. The curve, for which waiting time is plotted on the x axis as the independent variable, and number of users is plotted on the y axis as the dependent variable, showcases an increasing exponential fashion. This correlates well with the explanation of P2P networks presented in section 2.1.2 and the P2P theoretical logarithmic formula, in which number of users is the independent variable and waiting time is the dependent variable, giving a logarithmic trend inverse to the exponential. Similarly to CDNs, the theoretical curve is smooth while the experimental data collected increases in steps. This is due, again, to the fixed minimum time it takes for the users to download the complete file. Moreover, the experimental values match the theoretical prediction values for 1 segment, but as the number of segments increases the experimental values perform worse than the theoretical prediction. This is due to the fact that in practice, the most efficient transfer option is not always chosen and the P2P networks do not always send the segments through the most efficient route, which results in some nodes waiting for long periods of time to receive the segment.

2.3.3 Hybrid CDN-P2P Networks

The Hybrid CDN-P2P network was evaluated using the CDN-only edge server and the P2P-only configurations described in section 2.2.1 of this document. The results are examined on Figures 12 and 13.

It can be observed from Figure 12 that for a singular file segment the number of users that have received the full file at a given time increases linearly as the number of edge servers increases. However, the increase gradient for the experimental values is significantly lower than for the values predicted from the CDN-only theoretical model.

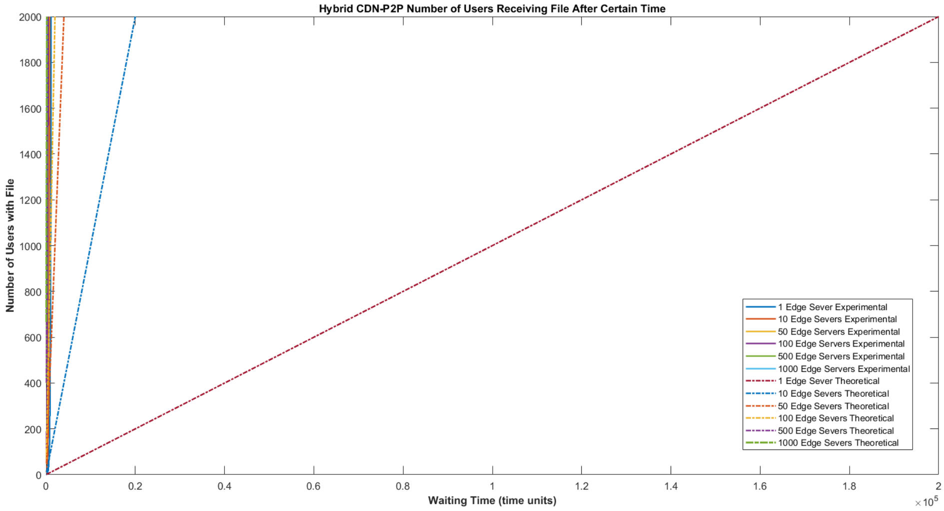

Figure 12: Hybrid CDN-P2P network graphical representation of the number of users with the file for different values of edge servers.

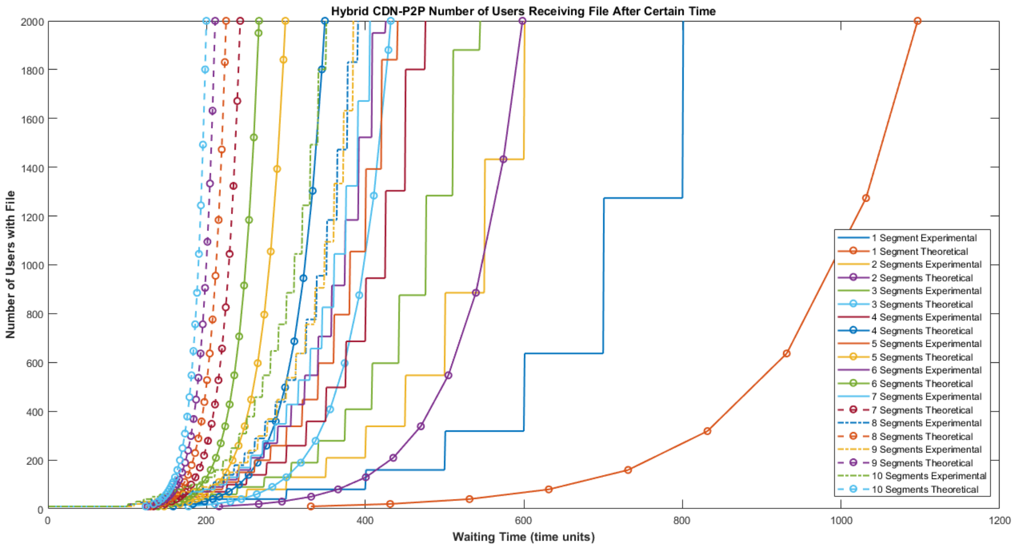

Figure 13: Hybrid CDN-P2P network graphical representation of the number of users with the file for different file segments.

From Figure 13 it can be seen that the number of users that have received the file at a given time increases as the number of file segments increases. The curve shape of the experimental data is similar to the P2P model theoretical predictions, but the experimental values are significantly lower. Overall, the Hybrid CDN-P2P network, although showing presenting a similar behaviour to the CDN-only and P2P-only networks is inaccurate compared to them. This decrease in performance could be due to a combination of factors. To being with, the network traffic load threshold that needs to be reached for the network to start using its P2P capability needs to be established correctly. An incorrect threshold will cause the network's performance to degrade. Moreover, the experimental trend increases in steps, as opposed to the theoretical curves which are smooth. The explanation for the stepping is analogous to the one previously presented for the CDN-only networks. Additionally, when using the P2P capability of the network and analogous to the problems presented by the P2P-only network, the most efficient routes and transfer options are not always implemented in practice. Finally, for an S-bit file and users with a download bandwidth of D bits/cycle. this network also presents a minimum delivery time of S/D seconds, following the same explanation provided for the CDN-only network.

2.4 Conclusion

From the analysis performed we can conclude that for CDN-only networks the number of users that have received a file after a certain time increases linearly as the number of edge servers or the upload bandwidth increases. In both cases, however, the waiting time for a user to receive the complete file stops decreasing in practice when a threshold is met, remaining constant.

For P2P-only networks, the number of users that have received a file after a certain time increases in a logarithmic fashion as the number of peers in the network increases. An additional way to decrease the waiting time for the users to receive the full file is to split it into equal segments which can be distributed by the peers of the network. In practice, the most optimal route is not always chosen for the segments, causing the network's performance to differ slightly from the theoretical prediction.

Finally, Hybrid CDN-P2P networks behave in a similar way to CDNs when delivering a file that is not segmented. On the other hand, if the file is segmented, the Hybrid network behaves similarly to P2P networks. Nonetheless, in practice these networks are inaccurate and not recommended.

Overall, the network chosen for a particular scenario depends on the requirements for the task that needs to be performed. For instance, if a file cannot be segmented, CDNs can be an appropriate approach, while if the file can be segmented P2P networks will provide a faster service. For a combination of segmented and non-segmented files, Hybrid CDN-P2P networks can be considered, however, their tendency to differ from predictions must be considered.

3 Collaborative Filtering

Nowadays, content recommendation systems are used by a large number of companies such as Youtube [15] and Netflix [10] to boost their sales by increasing the public's consumption of their products. Moreover, content recommendation allows companies to avoid redundant storage of data by diversifying the viewing of their content. There are three main methods used for content recommendation: Collaborative Filtering (CF), Content-Based and Hybrid methods.

Collaborative Filtering methods are based on using a person's behaviour, activities and preferences to predict the preferences of similar users. They do not require the system to have an in-depth understanding of the content being consumed. The data used by these methods can range from ratings and likes/dislikes to the user's purchase history and list of viewed items. This methodology is based on the assumption that users that liked similar items in the past will like similar items in the future.

3.1 Methodology

The data gathered for this section of the work was analysed using Matlab software [8] and stored using Microsoft Excel software [9] in order to efficiently extract the necessary measurements and apply the correct mathematical formulae. Analogous to the Content Delivery section of this work, the data was stored in the University of Surrey GitLab repository `https://gitlab.eps.surrey.ac.uk/ee00157/mediacasting_cwk`.

3.2 Weighted Slope One Method

One of the most commonly used methods for predicting the rating of an item is the Slope One algorithm. In order to apply the Slope One method, we consider, for instance, two items A and B bought by the same user. Suppose the user has rated item A with a rating R_A and item B with a rating R_B. If a new customer buys item A and rates it P_A, the rating given by this second user to item B will be given by $P_B = P_A + R_B - R_A$. For a database of users, a deviation matrix is computed according to Figure 14 to take into account all the possible pairs of items.

Algorithm 1 Computation of Weighted Slope One deviation matrix.

Input: U: set of all users, S: set of all items, R: set of all rated items

Output: dev: deviation matrix, $Freq$: co-rating frequency matrix

for all $u \in U$ **do**
  for all $s \in R_u$ **do**
    for all $s' \in R_u$ **do**
      $dev_{s,s'} \leftarrow dev_{s,s'} + (r_{u,s} - r_{u,s'})$
      $Freq_{s,s'} \leftarrow Freq_{s,s'} + 1$
for all $s \in S$ **do**
  for all $s' \in S$ **do**
    $dev_{s,s'} \leftarrow dev_{s,s'} / Freq_{s,s'}$

Figure 14: Computation of the Weighted Slope One deviation matrix. Source: [4]

Given that a user has bought a collection of items Q we can predict the rating that this user is likely to give to a new item by using the deviation matrix and the rating for the items that the user already bought. The predicted rating can be computed from the mathematical expression $p_{u,i} = \frac{\sum_{j \in C_i}(dev_{i,j} + r_{u,j})}{|C_i|}$, where $p_{u,i}$ represents the predicted rating given by user u to each item i, $dev_{i,j}$ denotes the corresponding element of the deviation matrix, $r_{u,j}$

indicates the rating given by user u to the other item j paired with item i, and $|C_i|$ represents the set of items rated by the user u that have been brought together with item i by another user, which is denoted by $C_i = \{j / j \in R_u, j \neq i, |S_{i,j}| > 0\}$ [4].

The prediction for items that rated together more frequently should be taken into account in a greater proportion than for items that are co-rated less frequently. This is known as the Weighted Slope One method, and the mathematical expression to compute the predicted rating is given by $p_{u,i} = \frac{\sum_{j \in R_u} (dev_{i,j} + r_{u,j})|S_{i,j}|}{\sum_{j \in R_u} |S_{i,j}|}$, where R_u represents the set of items rated by user u co-rated with item i by another user, and $|S_{i,j}|$ denotes the number of co-ratings for the particular item pair since $S_{i,j}$ corresponds to the set of all users who co-rated item i and j [4]. The output recommendation provided to the user is composed of the top-N items with the highest predicted scores. This is given by p_u, i where $i = 1, 2, 3, \ldots N$ [12]. The computation of the Weighted Slope One predictions can be found on Figure 15.

Algorithm 2 Weighted Slope One predictions.

Input: u: target user of the prediction, s: item to be predicted, R_u: set of all items rated by user u, dev: deviation matrix, $Freq$: co-rating frequency matrix

Output: $p_{u,s}$: prediction of item s for user u

for all $s' \in R_u$ **do**

  $p_{u,s} \leftarrow p_{u,s} + (r_{u,s'} + dev_{s',s}) \times Freq_{s',s}$

  $TotalWeights \leftarrow TotalWeights + Freq_{s',s}$

$p_{u,s} \leftarrow p_{u,s} / TotalWeights$

Figure 15: Computation of the Weighted Slope One prediction. Source: [4]

Further information on the complete methodology for calculating both the Weighted Slope One deviation matrix $dev_{i,j}$ and its predictions can be found on [4].

3.3 Minimum Number of Recommendations

In order to account for the minimum number of recommendations, an indicator function can be used. An indicator function outputs a value of 1 when an event happens and value of 0 when the event does not occur. It is often used to simplify notation and demonstrate probability theorems. The indicator function of a subset A of a set X is a function $\mathbb{1}\alpha : X \to \{0,1\}$ defined as $\mathbb{1}_\alpha(\mathrm{x}) := \begin{cases} 1 & \text{if } x \in \alpha \\ 0 & \text{if } x \notin \alpha \end{cases}$. For this particular scenario we will use the following indicator function: $\mathbb{1}[\,|\mathrm{S}_{i,j}| \geq \alpha] := \begin{cases} 1 & \text{if } |\mathrm{S}_{i,j}| \geq \alpha \\ 0 & \text{if } |\mathrm{S}_{i,j}| < \alpha \end{cases}$, where $|S_{i,j}|$ is the number of recommendations or co-ratings for a given pair of items i,j and α is the minimum number of recommendations specified for the system.

The indicator function can be incorporated into the Weighted Slope One Method by multiplying it by both the numerator and the denominator inside the summations of the mathematical expression outlined in the previous section of this document to calculate the predicted rating $p_{u,i} = \frac{\sum_{j \in R_u} ((dev_{i,j} + r_{u,j})|S_{i,j}| \mathbb{1}[|S_{i,j}| \geq \alpha])}{\sum_{j \in R_u} (|S_{i,j}| \mathbb{1}[|S_{i,j}| \geq \alpha])}$. In this expression, $p_{u,i}$ represents the predicted rating given by user u to each item i, $dev_{i,j}$ denotes the corresponding element of the deviation matrix, $r_{u,j}$ indicates the rating given by user u to the item paired with item i, R_u denotes the set of items rated by user u co-rated with item i by another user, $|S_{i,j}|$ indicates the number of co-ratings for the i,j item pair, and $\mathbb{1}_\alpha(x)$ is the indicator function. If the number of recommendations $|S_{i,j}|$ is greater than the threshold α set as the minimum number of recommendations, the indicator function will output 1, otherwise it will output 0 and the particular i,j item pair will not be added to the calculations. Given a situation in which none of item pairs meet the recommendations threshold, both the numerator and the denominator of the full expression will equal 0. This results in a mathematical indeterminacy, from which we can assume that the prediction does not exist. This is indicated as a prediction value of -1 in the practical data collected during the laboratory session.

3.4 Root Mean Square Error (RMSE)

In order to evaluate the accuracy of a content recommendation system, the Root Mean Square Error (RMSE) between the predicted ratings and the real i.e ground truth ratings was computed. For a group of users and movies T, the RMSE can be computed by using the mathematical expression $RMSE = \sqrt{\frac{\sum_{(u,s)\in T}(p_{u,s} - r_{u,s})^2}{|T|}}$, where u indicated the user, s the specific movie, p the predicted rating score for the movie by user u, and r the ground truth rating for the movie.

Additional information on the performance of the system can be obtained by calculating both the Precision and the Recall values of the system. For a given system, the precision value is given by the number of returned results that are relevant, and the recall value is given by the number of relevant results that have been returned. An F1 curve combining both Precision and Recall can then computed for evaluation.

3.5 Experimental Data and System Performance Analysis

For this particular experiment, a subset of the MovieLens dataset was used as a dataset. This subset includes 100,000 ratings for approximately 1,700 movies viewed by around 1,000 users. The dataset was split into four train-test groups. For each group 80% of the data was used as training data and the remaining 20% of the data was used as test data, making this a supervised classification problem.

The program used takes as inputs a training file, a test file, and output file name and a number of minimum required recommendations. It then uses the data in the training and test files to compute the deviation and frequency matrices, which are used to predict the ratings given by different users to different movies. If a pair of movies is not co-rated at least the number of times specified by the minimum recommendation value provided, a prediction is not performed on that particular pair.

For each train-split cohort u1, u2, u3 and u4 an RMSE value was calculated for each minimum recommendation value of 10, 20, 30, 40, 50, 60, 70, 80, 90 and 100, with the results without a prediction being discarded. In order to interpret the information the average RMSE of the four systems was computed for each minimum recommendation value. Table 12 shows the RMSE value for each minimum recommendation value in each system, as well as the total RMSE values, and Figure 16 provides a visual representation of the values.

	RMSE				
Minimum Recommendation	**RMSE U1**	**RMSE U2**	**RMSE U3**	**RMSE U4**	**Total RMSE**
10	1.061	1.039	1.023	1.029	1.038
20	1.062	1.045	1.022	1.032	1.040
30	1.060	1.044	1.028	1.034	1.042
40	1.062	1.041	1.034	1.033	1.043
50	1.063	1.038	1.034	1.029	1.041
60	1.070	1.041	1.035	1.034	1.045
70	1.063	1.049	1.039	1.037	1.047
80	1.061	1.035	1.040	1.039	1.044
90	1.060	1.042	1.040	1.030	1.043
100	1.065	1.041	1.056	1.045	1.051

Table 12: RMSE values for the different systems at different thresholds for the minimum number of recommendations.

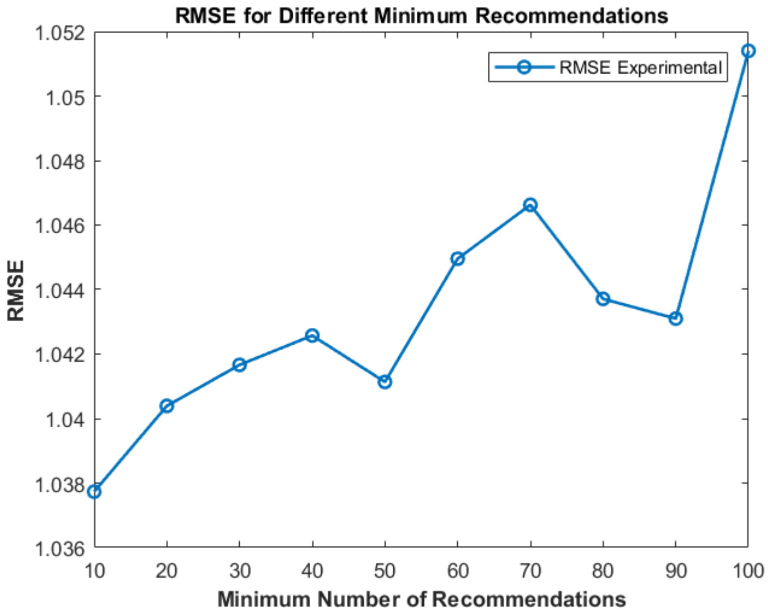

Figure 16: Graphical representation of the total RMSE values for the different minimum number of recommendations threshold values.

From the Figure 16 it can be observed that as the minimum number of recommendations increases, the total RMSE of the system follows an overall increasing trend. Therefore, the performance of the system overall decreases as the minimum number of recommendations increases. This is expected, as the greater the number of recommendations considered is, the harder it will be for the system to find item pairs with such a high number of co-ratings to calculate the predictions. Consequently, since there is less data available to compute the prediction, the predicted rating is likely to be less accurate.

It can be concluded that the present system is not robust to the minimum number of recommendations required since its performance varies depending on this parameter. A robust system should perform similarly for different minimum numbers of recommendations, and this system does not fulfill that criteria.

3.6 Top-K Recommendations Accuracy Analysis

3.6.1 Analysis Procedure

In order to analyse the accuracy of the top-K recommendations provided by the system i.e the percentage of the K movies recommended by the system that a specific user actually likes, a ground truth rating value of 4 or above was considered equivalent to a user enjoying a movie. Both the predicted and the ground truth values were extracted for each user in each train-test split for each minimum recommendation value, with the data for that specific user being sorted in descending order for the predicted rating values. A comparison between the ground scores and the predicted scores for that user was then performed on the top K movies to determine whether the user liked them or not. From this comparison, the percentage of movies that the user likes was computed by using the expression $Percentage = 100 * \frac{moviesLiked}{K}$. This analysis was performed for different values for K being an integer that ranges from 1 to 20 for each user independently, giving one curve per minimum recommendation value (10, 20, 30, 40, 50, 60, 70, 80, 90 and 100) for each dataset (u1, u2, u3, u4). Averaging these graphs results in an average plot for each minimum recommendation value [12].

Additional measurements to further evaluate the system's performance were incorporated as an extension to this section of the work. These measures are the Precision, Recall and F1 of the system for the top-K recommendations, where K ranges from 1 to 20. The Precision of the system is given by $P = \frac{NumberOfRelevantItemsRetrieved}{TotalNumberOfRetrievedItems}$, the Recall of the system is represented by $R = \frac{NumberOfRelevantItemsRetrieved}{TotalNumberOfRelevantItems}$. F1 is a measure that allows to examine the Precision and Recall of the system simultaneously by using $F1 = \frac{2*Precision*Recall}{Precision+Recall}$. These measures can be represented as a percentage by multiplying the result of the fraction by 100. It can be seen that the Precision for the top-K recommendations is equivalent to the above-mentioned calculation of the top-K recommendations accuracy. Therefore, the analysis of the system's accuracy was performed by comparing the Top-K Accuracy, Recall and F1 graphs.

3.6.2 Analysis Results

The percentage of correct predictions for different values of K was computed for each dataset, producing one graph per train-test set. The graphs for the sets u1, u2, u3 and u4 were averaged using the standard arithmetic mean to produce an average plot for the correct predictions for every minimum recommendation value. The same averaging procedure was implemented to compute the Recall and F1 percentages. The results can be found on Figures 17 through 19.

It can be seen from Figure 17 that the top-K accuracy of the system generally increases for an increasing number of minimum recommendations. This increase in accuracy is expected due to the fact that by imposing a high number of minimum recommendations on the system and sorting each user's data in decreasing predicted rating order, there is a higher chance of obtaining a correct - with a ground truth rating equal to or higher than 4 - top item the more strict the filtering system is i.e the higher the threshold for the minimum number of recommendations is. Exceptions to this are the accuracy percentages obtained for 10 and 20 minimum recommendations. Both of these top-K accuracy measurements are overall greater than those obtained for 30, 40 and other minimum recommendation values. A possible explanation for this would be the fact that there is a large number of users selected for these two categories, therefore incorrect predictions for a few of them are negligible and balanced out by the average obtained from the correct predictions in the larger cohort, which will not be the case for intermediate minimum recommendation values such as 40, 50, 60 and others.

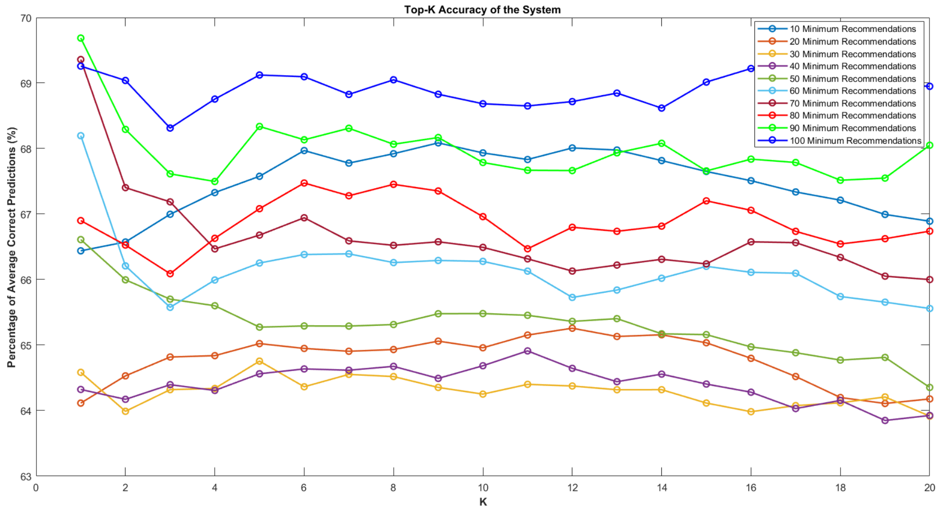

Figure 17: Graphical representation of the top-K accuracy of the system for different values of minimum recommendations.

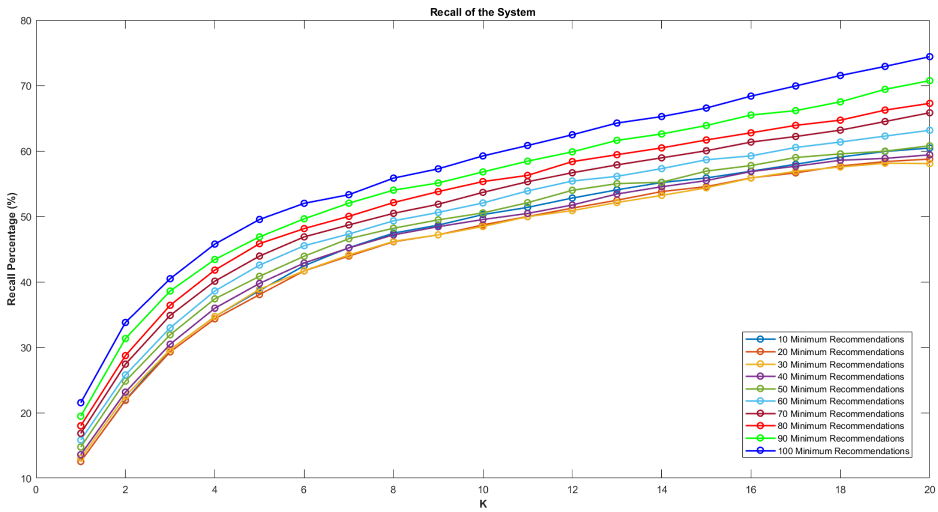

Figure 18: Graphical representation of the recall percentage of the system for different values of minimum recommendations.

It can be observed that, generally, as K increases the top-K accuracy decreases. This is especially true for high values of minimum number of recommendations such as 40, 50 and 60, where the curve shows a steep decrease from a K value of 1 to a K value of 4, which correlates well with the expectation. Since the items with higher rating are situated at the top of the sorted list, as more items are considered i.e as K increases, the predicted ratings for the items selected are lower, which increases the chance of these items being incorrect predictions.

The curves for the different number of minimum recommendations do not show the exact same shape. Even though, overall, they all present similar gradients, they show different jumps and increasing or decreasing tendencies between the points. For instance, while a value of 90 minimum recommendations decreases in accuracy between the K values of 1 and 3, a value of 30 minimum recommendation increases in accuracy between those same values, yet both curves can be approximated to a straight horizontal line.

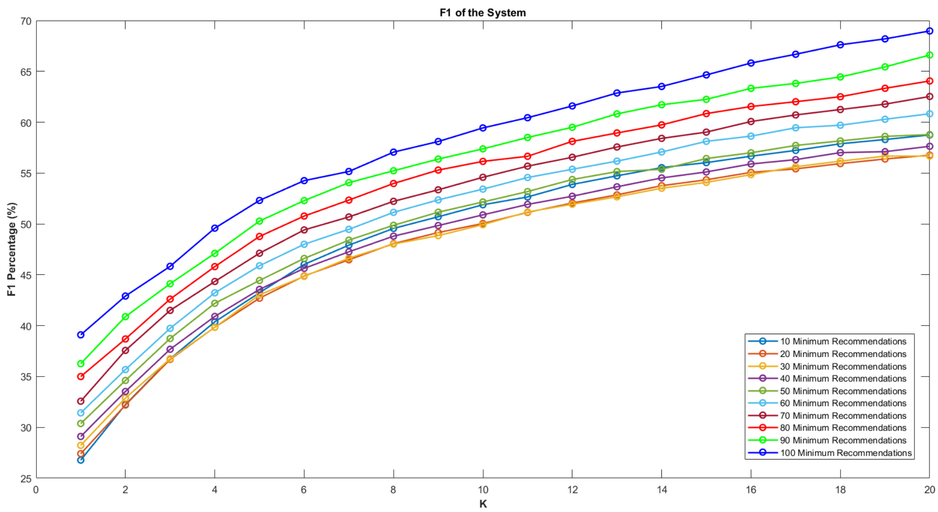

Figure 19: Graphical representation of the F1 percentage of the system for different values of minimum recommendations.

Regarding the Recall measure of the system, as observed in Figure 18, for all the values of minimum recommendations the recall percentage for a K value of 1 and increases in a near-logarithmic trend until a K value of 20. The trend followed by the F1 measure of the system varies in the same way as the Recall measure albeit consisting of higher initial percentage values and lower final accuracy values which lead to a less sharper gradient of the curve overall. Thus, the trend of the totality of these curves can be categorised as an increasing near-logarithmic function.

From the top-K accuracy percentages, it can be deduced that for the minimum recommendation values of 10 and 80 the optimal K value is 6, and for a minimum recommendation value of 20 the optimal K value is 12. For 30 minimum recommendations, a K value of 5 gives the highest accuracy, while for a minimum recommendation of 40 it is a K value of 11 that performs best. Additionally, for minimum recommendation thresholds of 50, 60, 70 and 90 the optimal K value is 1. Finally, for 100 minimum recommendations, it is a K value of 16 that produces the highest outputs. While in the cases of 50, 60, 70 and 90 minimum recommendations the difference in performance between a K value of 1 and other values is significant, for other curves such as 30 or 40 minimum recommendations, the performance is comparable for all K values and presents less variation. Nonetheless, when taking not only the top-K Precision but also the Recall measure of the system through the F1 measure on Figure 19, the optimal K values vary, since we consider all the relevant items as opposed to only the retrieved ones. Thus, the value of K should not be chosen independently, but instead taking into account the purpose of the system and whether its Precision or Recall performance is of a greater importance.

3.7 Conclusion

From this analysis it can be seen that the Weighted Slope One method is a simple yet efficient way of predicting the rating given by a particular user to an item based on both the ratings given by that user to other items and the ratings given by similar users to that item.
The accuracy of the Weighted Slope One system was analysed by calculating the Root Mean Square Error (RMSE), the accuracy of the top-K recommendations for different integer values

of K from 1 to 20, and the Recall and F1 values for the various thresholds for the minimum number of recommendations.

The RMSE results show that as the threshold set for the number of minimum recommendations is increased, the Root Mean Square Error (RMSE) of the system increases, hence the performance of the system degrades. Regarding the top-K accuracy of the system, for higher number of minimum recommendations the top-K accuracy of the system is generally higher, and for high values of K the accuracy tends to decrease. This correlates well with the fact that the higher the minimum number of recommendations threshold, the more likely the top item is to be a correct prediction, as well as with the fact that a higher minimum recommendations threshold prevents the system from outputting enough recommended items for a many of the users to satisfy a K value of 18, 19, 20 or similar.

Overall, it can be concluded that the system's accuracy satisfies the theoretical predictions for the performance of the Weighted Slope One method. Further work to analyse the system would require implementing comparisons between the Weighted Slope One technique and other collaborative filtering methods such as the Amazon Item-Item Collaborative Filtering method on the given databases.

REFERENCES

[1] Akamai. Akamai, 2019. Available: "`https://www.akamai.com/uk/en/`" [Accessed 10-03-2018].

[2] Apple. iTunes, 2019. Available: "`https://www.apple.com/uk/itunes/`" [Accessed 13-03-2018].

[3] BBC. iPlayer, 2019. Available: "`https://www.bbc.co.uk/iplayer`" [Accessed 13-03-2018].

[4] L. Silva et al. D. Menezes, A. Lacerda. Weighted slope one predictors revisited. *International World Wide Web (WWW) Conference*, May 2013.

[5] E. Quacchio et al. E. Baccaglini, M. Grangetto. A study of an hybrid cdn–p2p system over the planetlab network. *Signal Processing: Image Communication*, Feb 2012.

[6] Nemeth Evi. *UNIX and Linux system administration handbook*. Boston: Pearson Education, 2018.

[7] Linux. Linux, 2019. Available: "`https://www.linux.org/`" [Accessed 11-03-2018].

[8] Matlab. Mathworks, 2019. Available: "`https://www.mathworks.com/products/matlab.html`" [Accessed 11-03-2018].

[9] MicrosoftExcel. Microsoft, 2019. Available: "`https://products.office.com/en-gb/excel`" [Accessed 11-03-2018].

[10] Netflix. Netflix, 2019. Available: "`https://www.netflix.com/gb/`" [Accessed 10-03-2018].

[11] Dr Eng-Jon Ong. *EEE3034 MEDIACASTING: Content Delivery Networks Lecture notes*. University of Surrey, 2019.

[12] Dr Eng-Jon Ong. *EEE3034 MEDIACASTING: Content Recommendation Lecture notes*. University of Surrey, 2019.

[13] B. Akbari S. M. Y. Seyyedi. Hybrid cdn-p2p architectures for live video streaming: Comparative study of connected and unconnected meshes. *2011 International Symposium on Computer Networks and Distributed Systems (CNDS)*, Feb 2011.

[14] University of Surrey. GitLab, 2019. Available: "`https://gitlab.eps.surrey.ac.uk/`" [Accessed 14-03-2018].

[15] YouTube. YouTube, 2019. Available: "`https://www.youtube.com/`" [Accessed 10-03-2018].