SAND2011-2879P

Peer to Peer vs. Client Server File Sharing Scenarios

Nathan Spencer
August 30, 2007

Executive Summary

Peer to peer file sharing strategies have demonstrated an efficient process of distributing large files over the internet to many users. Rather than relying on a dedicated server to upload files to clients, peer to peer networks allow downloaders to become uploaders to other users requesting the file. In this manner, an exponential increase in uploaders is possible, and often observed immediately after a popular, new file is introduced on the internet [1]. The net effect of this increase in uploaders is that the time to download a file from the network usually becomes dependent on the downloading bandwidth (rate of file transfer in bits per second) of the requester rather than the uploading constraints of a server. Typically the uploading constraint of a server is the bottle neck in a file transfer process, especially when a large number of clients are attempting to download a file from a server simultaneously. The advantage of peer to peer networks in distributing a high demand file to many users is well accepted, both theoretically and by observation of internet traffic [1-3]. However, the performance advantage begins to decline as the demand for a file decreases. In the extreme case of many files being requested by a single client (transfer for storage or single user analysis), there is no improvement in the transfer time or bandwidth requirements when using a peer to peer file sharing strategy over a client server connection.

Other key insights of the performance of peer to peer networks are summarized herein. They are based on results from a mathematical model used to predict the outcomes of three different scenarios. These scenarios are chosen for possible operational relevance and to demonstrate the performance bounds of the file sharing options, but are not specific to a particular application. The scenarios are as follows:

1) **A sudden explosion of requests is made for a particular file.** This may occur in an emergency situation, when situational awareness is desired instantaneously at thousands of locations. This is an extreme example of a single file in high demand.
2) **A steady flow of requests are made for a particular file.** This scenario may be realized in situations where popular files may be distributed to requesters in a subscription manner. This scenario demonstrates the point at which peer to peer networks first become advantageous.
3) **Files are transferred from a point of origin to a single destination.** The acquisition of large data files in field operations which are transferred to a single analysis or data storage facility is categorized by this scenario. As mentioned previously, there is no improvement in transfer time or bandwidth requirements when using a peer to peer network in this situation.

The method of mathematically modeling these scenarios is based largely on the work presented by Qiu and Srikant [2]. The implementation of the mathematical model is checked by comparing its results with those reported in the literature. The model is then modified to adapt it for the specific scenarios. Some of the insights acquired from this process are summarized below. The remaining sections of this report include the details of these conclusions.

When is a peer to peer file sharing strategy advantageous over a client server one?

When the file transfer is upload constrained and there is more than one downloading request. Bandwidth is most commonly upload constrained, i.e. the rate of transfer of files to the downloader is usually higher than that from an uploader to the internet. If there is only one requester during the life of the file, then there is no demand for the new acquirer to share it with another and the peer to peer capability will not be utilized. The greatest performance advantage comes when files are large and there are many downloading requests. In this regards, peer to peer networks are considered to be *scalable,* which is to say that at steady state the time required to download a file is not dependent on the number of new requests [2].

How long does it take to download a file in a peer to peer network?

A peer to peer network may be either download or upload constrained (with a slight chance of the rates being equal). If it is download constrained, it will remain so and the time to acquire the file is simply determined by the downloading bandwidth and the file size (neglecting any latency characteristics). If it is upload constrained, the transfer time is also determined by the file size and uploading bandwidth. However, the uploading bandwidth improves as downloaders also become uploaders. As more uploaders are available, the network may evolve into being download constrained.

Does compressing files aid in the file transfer process?

Compressing a file will improve the time required to transfer it for a given bandwidth because of its smaller size. The tradeoff between the computational cost associated with compressing a file and the improved file transfer time is dependent on the file type and the compression method. In this study, the effects of compression are accounted for by simply considering files of different sizes.

How does splitting up the file into smaller parts help in the peer to peer transfer process?

When a large file is split up into smaller parts, the smaller parts may be shared with other uploaders sooner and in a more efficient manner. The efficiency of file sharing among downloaders increases rapidly as the file is initially split into parts, but returns diminish quickly as the number of parts increase.

Table of Contents

List of Figures

List of Tables

Peer to Peer BitTorrent Based Deterministic Flow Model

The peer to peer deterministic flow model presented in this section is based on the work of Qiu and Srikant [2]. It was created to study the performance of BitTorrent, a specific file sharing program. The authors also compared the model predictions to that of an actual test where a file was introduced and distributed on the internet via BitTorrent. The test results contained significant oscillations which are not captured by a smooth, deterministic model. However, the dynamic insights from the model are instructive and a basis for comparing performance metrics between peer to peer and client server networks.

The implementation of the model is tested by comparing its results with those reported in the literature. In the next section, the model is modified for adaptation to the specific scenarios under consideration.

Model Description

The model is developed by considering two types of peers: those who are downloaders and those who are seeds. The seeds are those who have a complete copy of the desired file and remain online to allow others to download it. Downloaders are those who do not yet have the complete copy of the file and are currently acquiring it, but may also upload parts of the file to other peers. Downloaders enter the system at a rate indicated as new requests, and may leave the system by aborting the download process. They become seeds after completing the download. Seeds leave the system at a determined rate. This process is shown in the flow diagram in Figure 1.

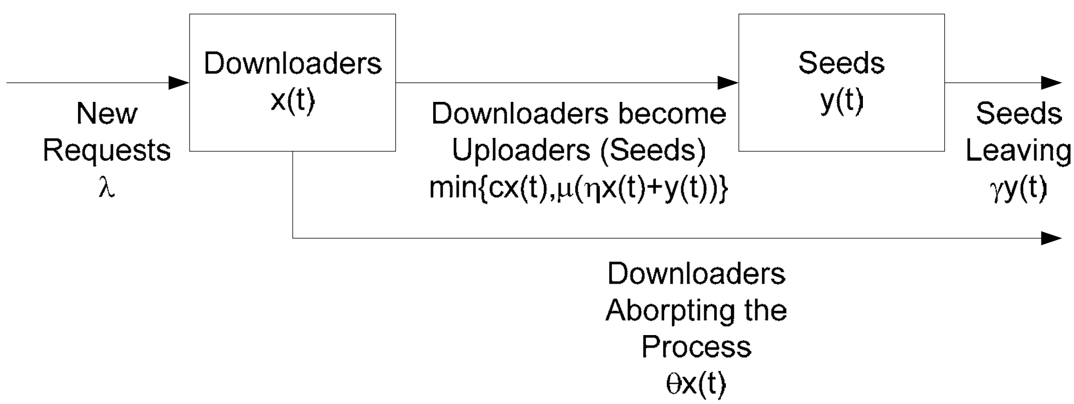

Figure 1 Systems diagram of the peer to peer network deterministic flow model.

The mathematical equations representing the deterministic flow process and variable definitions are as follows:

$$\frac{dx}{dt} = \lambda - \theta x(t) - \min\{cx(t), \mu(\eta x(t) + y(t))\}$$

$$\frac{dy}{dt} = \min\{cx(t), \mu(\eta x(t) + y(t))\} - \gamma y(t)$$

Subject to: $x, y, c, \lambda, \mu, \eta, \gamma, \theta \geq 0$

Table 1 Variable descriptions and baseline values for comparison with results in the literature.

Variable	Description	Baseline Value	Units
x(t)	Number of downloaders in the system at time t	0 at t=0	downloaders
y(t)	Number of seeds in the system at time t	1 at t=0	seeds
λ	Arrival rate of new requests (downloaders)	1	downloaders/min
μ	Uploading bandwidth of a given peer	0.00125	files / min
c	Downloading bandwidth of a given peer	0.002	files / min
θ	Rate of downloaders aborting the download	0.001	downloaders / min
γ	Rate at which seeds leave the system	0.001	seeds / min
η	Effectiveness of downloaders also uploading to other downloaders, interval [0,1]	0 for x(t)<1 1 for x(t)>1	unitless

The fact that the transfer process may be either upload or download constrained is captured in the selection of the minimum of two rates. The first, *cx(t)*, is the download constraint. The second considers the upload constraint based on the total number of peers capable of uploading to each other and the upload bandwidth. The parameter η is used to capture the effectiveness of downloaders downloading to each other. It can be shown that η is related to the number of parts that the file is split into, *N*, and the total number of peers that a downloader can download from, *K*, by the following relationship:

$$\eta \approx 1 - \left(\frac{\log N}{N}\right)^k$$

Where $k = \min\{x\text{-}1, K\}$. From observation, η is zero for *N*=1 and quickly approaches 1 for even small *N* and *k* (e.g. *h* = ~0.98 for *N,k* = 2). The maximum value of η is 1, so further increases in *N* and *k* have little effect on η.

Underlying Assumptions in the Model

Some of the underlying assumptions in the above model are stated explicitly as follows:

1) The network bandwidth is not a constraint. The network is capable of handling any volume of data.
2) All peers have the same upload and download bandwidths.
3) One file of constant size is being transferred.
4) Downloaders may instantly become uploaders to others.

Steady State Performance

Steady state performance can be determined by setting the time derivatives equal to 0 and solving for x and y for both upload and download constrained cases. An eigenvalue analysis also shows that these points are locally stable [2].

Download Constrained, $c\bar{x} \le \mu(\eta\bar{x} + \bar{y})$:

$$\bar{x} = \frac{\lambda}{c\left(1 + \frac{\theta}{c}\right)} \qquad \bar{y} = \frac{\lambda}{\gamma\left(1 + \frac{\theta}{c}\right)}$$

Upload Constrained, $c\bar{x} \ge \mu(\eta\bar{x} + \bar{y})$:

$$\bar{x} = \frac{\lambda}{\nu\left(1 + \frac{\theta}{\nu}\right)} \qquad \bar{y} = \frac{\lambda}{\gamma\left(1 + \frac{\theta}{\nu}\right)}$$

Where

$$\nu = \frac{\eta\mu\gamma}{\gamma - \mu} \Rightarrow \frac{1}{\nu} = \frac{1}{\eta}\left(\frac{1}{\mu} - \frac{1}{\gamma}\right)$$

Baseline Comparison to the Literature Results

Visual examinations of the results of the model are compared with those reported in the literature [2] by comparing plot trends and magnitudes. A summary of this comparison is in Table 2. Based on this comparison and the expected equilibrium values, the code implementation appears to be a correct representation of the mathematical model.

Table 2 Comparison of model results with those reported in the literature.

Test Case	Input Parameters	Comments
Download Constrained	Baseline Values	Trends and steady state values appear consistent.
Upload Constrained	Baseline Values, with $\gamma = 0.005$	Trends appear consistent. The steady state values for the number of seeds are consistent. However, the values for the number of downloaders deviate. The calculated steady state number of downloaders from the equilibrium analysis matches those reported herein.
Experimental Case	Baseline Values, with $\mu = 0.0013$ time dep. λ	Trends and steady state values are consistent. Peak values for both downloaders and uploaders appear to be lower than the literature values.

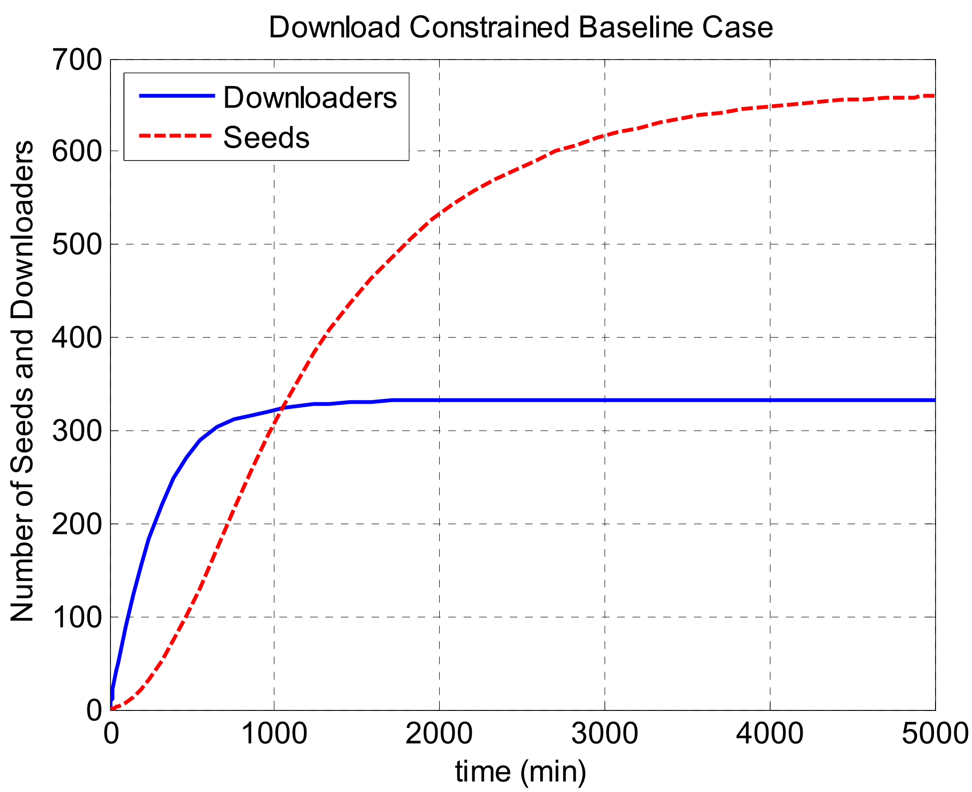

Figure 2 Plot of the download constrained baseline case for comparison to Qiu [2].

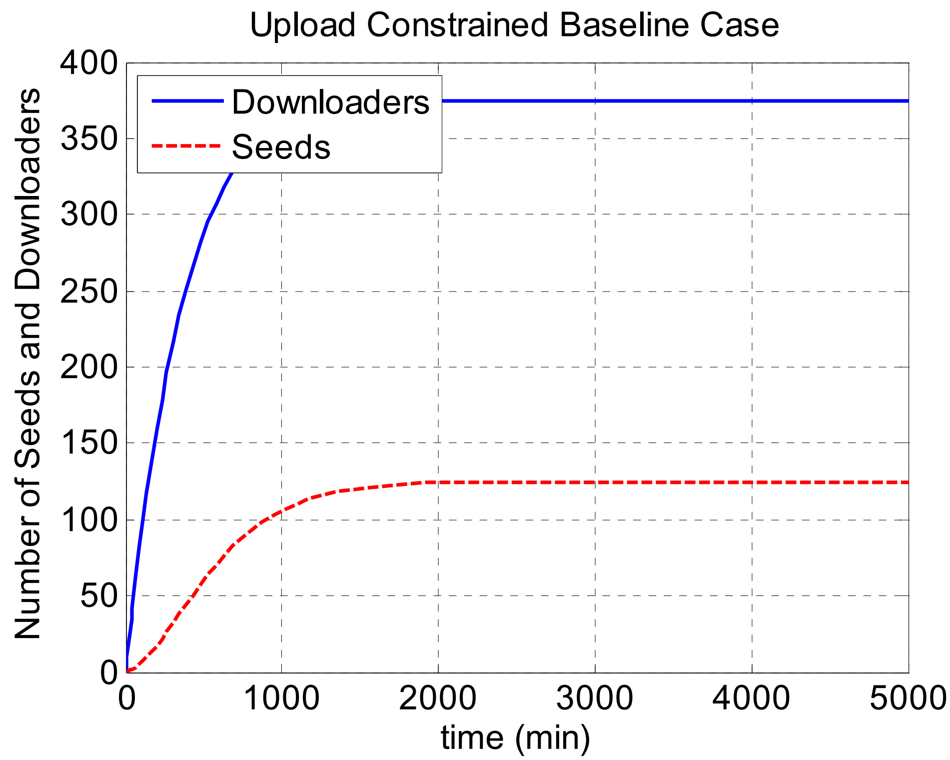

Figure 3 Plot of the upload constrained baseline case for comparison to Qiu [2].

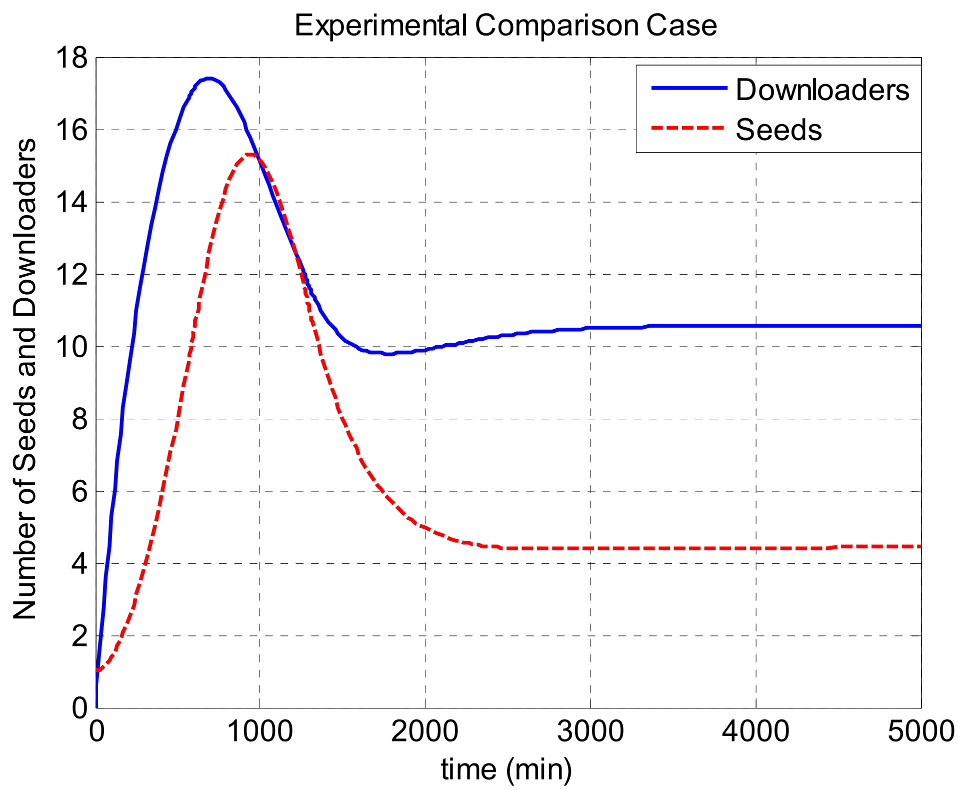

Figure 4 Plot of the experimental case for comparison to Qiu [2].

Model Adaptation for Specific Scenarios

The model was adapted for the specific scenarios in the following ways:

1) **The option for seeds and downloaders to leave the system is removed.** This is done to consider a situation in which policy is enforced to keep seeds active and not allowing downloaders to abort the process. This enforcement is only reasonable for an organization to enforce such policy, such as a corporation or government entity. The effect on this change is that the peer to peer network will perform more efficiently.
2) **The number of peers in the system is finite.** Again, this is done to model performance of a group of computers in a corporate or government entity. Steady state equilibrium values will be affected. An unstable equilibrium in the number of downloaders will be reached as a peak in downloaders, followed by a decrease to zero as all downloaders complete their transfer and are considered as seeds. The number of seeds simply increases until all peers are seeds.
3) **A finite time exists between when a peer starts to download and when it can start to upload to another peer.** This finite time interval is the time it takes to download the first part of a file before it can upload it to another peer. An introduction of a third state variable, the number of requests, is introduced to enable this process.
4) **A client server option is included for comparison.** The client server model has the same structure as the peer to peer model, except the number of seeds is set as a constant, equal to the number of servers available.

A diagram for the modified peer to peer model is shown in Figure 5.

Peer to Peer

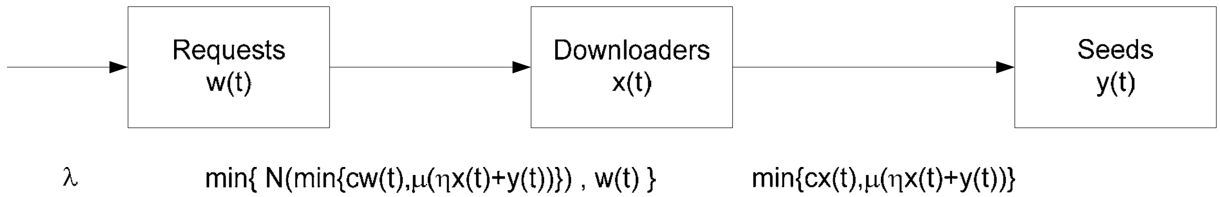

Figure 5 Diagram of the modified peer to peer model.

The mathematical equations representing the updated flow process are:

$$\frac{dw}{dt} = \lambda(t) - \min\{N \min\{cw(t), \mu(\eta x(t) + y(t))\}, w(t)\}$$
$$\frac{dx}{dt} = \min\{N \min\{cw(t), \mu(\eta x(t) + y(t))\}, w(t)\} - \min\{cx(t), \mu(\eta x(t) + y(t))\}$$
$$\frac{dy}{dt} = \min\{cx(t), \mu(\eta x(t) + y(t))\}$$

Subject to: $w, x, y, c, \lambda, \mu, \eta, \gamma, \theta \geq 0$
$N \geq 1$

The expression describing the transition from requesters to downloaders is formed by considering the time it takes to download the first part of a file separated into N parts. The units of bandwidth used in the model are in files per unit time. Therefore, the transfer rate to complete a Nth part of a file is N times the bandwidth. This bandwidth is the minimum of the uploading or downloading bandwidths. With a finite number of peers in the network, there is a possibility that the number of remaining requesting peers is less than the bandwidth capability to serve new peers, hence, the second minimum function.

The client server network is modeled using the same structure with a few more modifications. The flow diagram is shown in Figure 6. Instead of being called seeds, the clients which have completed the downloading process are simply called owners because they cannot upload to any other users. The number of servers, *S*, which upload the files remains constant, so the rate of change will remain constant until all requests are fulfilled.

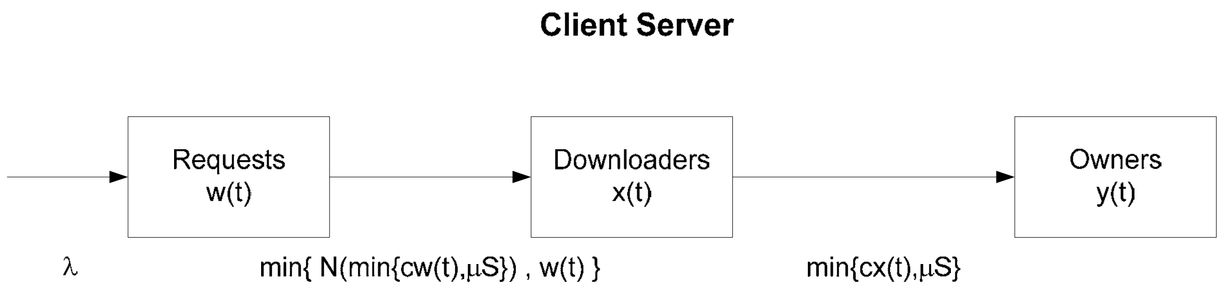

Figure 6 Diagram of the client server model.

The system equations are:

$$\frac{dw}{dt} = \lambda(t) - \min\{N \min\{cw(t), \mu S\}, w(t)\}$$
$$\frac{dx}{dt} = \min\{N \min\{cw(t), \mu S\}, w(t)\} - \min\{cx(t), \mu S\}$$
$$\frac{dy}{dt} = \min\{cx(t), \mu S\}$$

Subject to: $w, x, y, c, S, \lambda, \mu \geq 0$

$N \geq 1$

Scenarios

Three different scenarios are selected based on possible situations which result in different performance trends. The scenarios are general in nature, and are not designed for a particular application. The resulting trends demonstrate some of the dynamics of the file sharing systems, such as when peer to peer networks first become advantageous over client server networks and how much benefit is gained. In order to define a performance advantage, a metric is chosen as the time required to distribute a file to a selected number of peers. The scenarios are summarized as follows:

1) **A sudden explosion of requests is made for a particular file.** This may occur in an emergency situation, when situational awareness is desired instantaneously at thousands of locations. This is an extreme example of a single file in high demand.
2) **A steady flow of requests are made for a particular file.** This scenario may be realized in situations where popular files may be distributed to requesters in a subscription manner. This scenario demonstrates the point at which peer to peer networks first become advantageous.
3) **Files are transferred from a point of origin to a single destination.** The acquisition of large data files in field operations which are transferred to a single analysis or data storage facility is categorized by this scenario. As mentioned previously, there is no improvement in transfer time or bandwidth requirements when using a peer to peer network in this situation.

Scenario 1

A sudden explosion of requests is modeled by setting the initial number of requests to 1000. The baseline values for this and the remaining parameters for both peer to peer and client server models are summarized in Table 3. The upload and download bandwidths, μ and c, are calculated based on a file size of 860 kB and an uploading capacity of 0.65 Mbps and a downloading capacity of 1 Mbps. These values are selected based on the size of a compressed file of interest and advertised broadband capacity of a wireless internet service provider. The progression of the number of requests, downloaders, and owners of the file are shown in Figure 7 for the peer to peer network, and in Figure 8 for the client server network. The exponential change in requests to owners for the peer to peer case versus the linear nature of the client server network is a salient difference between the two. As a result, the time required to distribute the files is about 180 times faster for the peer to peer network. A direct comparison of the two methods is shown in Figure 9, by comparing the number of file owners as a function of time.

The dependency of the time required to complete the 1,000 file transfers on the file size, number of file divisions, and the bandwidths is investigated by varying each of this parameters independently. The effect of varying the file size ranging from 1 kB to 1 GB is shown in Figure 10 (latency effects are not included). Likewise, N is varied from 1 to 15 and the results are shown in Figure 11. The bandwidths, μ and c, are uniformly scaled up to a factor of 10, and an inversely proportional relationship is shown in Figures 12 and 13 (e.g. twice the bandwidth requires about half the transfer time).

Table 3 Scenario 1 baseline parameters.

Variable	Description	Baseline Value	Units
w(t)	Number of requests at time t	1000 at t=0	requesters
x(t)	Number of downloaders in the system at time t	0 at t=0	downloaders
y(t)	Number of seeds in the system at time t	1 at t=0	seeds
λ	Arrival rate of new requests (downloaders)	0	downloaders/sec
μ	Uploading bandwidth of a given peer	0.0945	files / sec
c	Downloading bandwidth of a given peer	0.1453	files / sec
η	Effectiveness of downloaders also uploading to other downloaders, interval [0,1]	0 for x(t)<1 1 for x(t)>1	unitless
N	Number of file divisions	4	unitless
S	Number of servers	1	servers

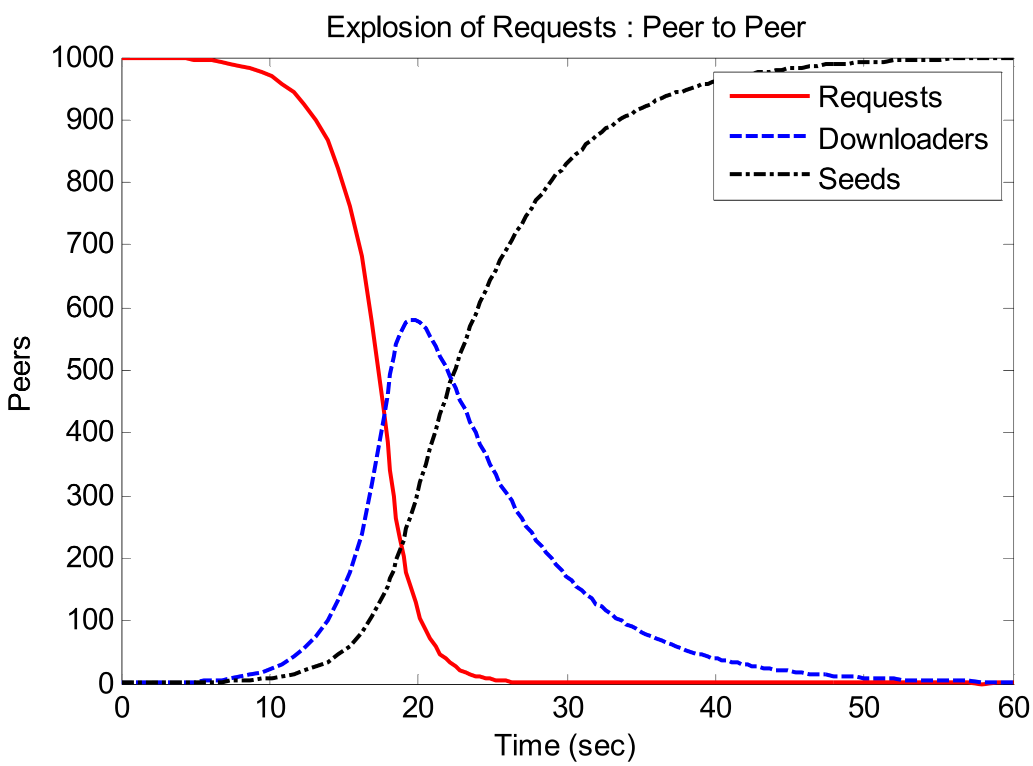

Figure 7 Progression of the state variables for a peer to peer network in scenario 1.

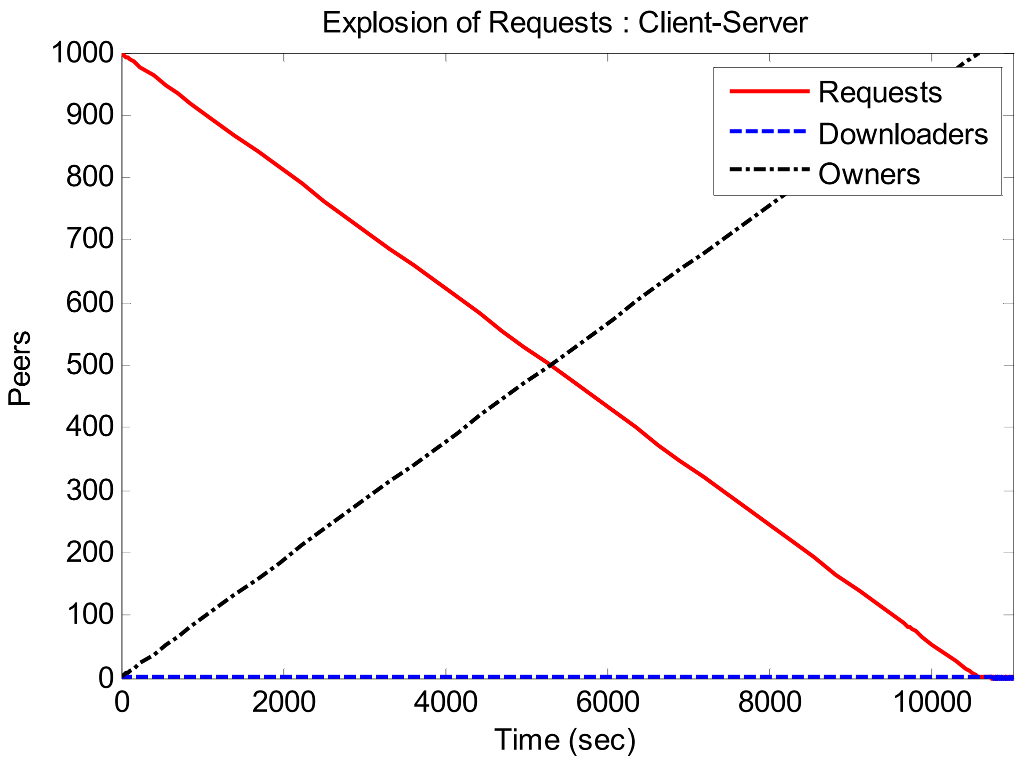

Figure 8 The progression of the state variables a client server network for scenario 1.

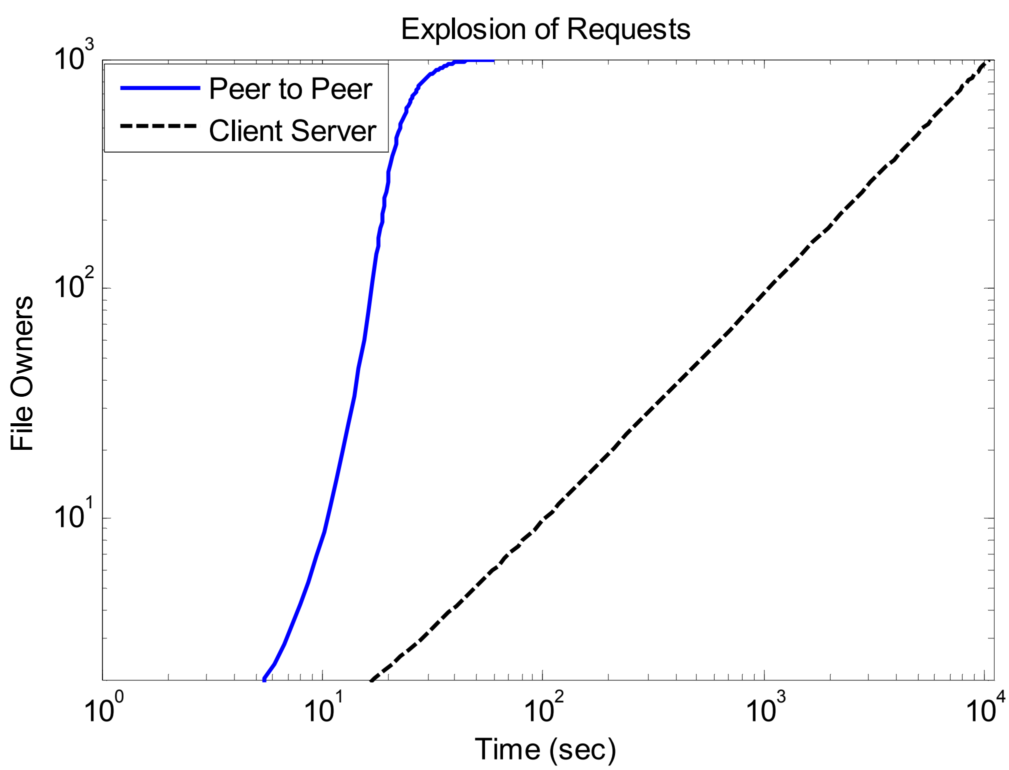

Figure 9 Peer to peer and client server comparison of the number of file owners as a function of time.

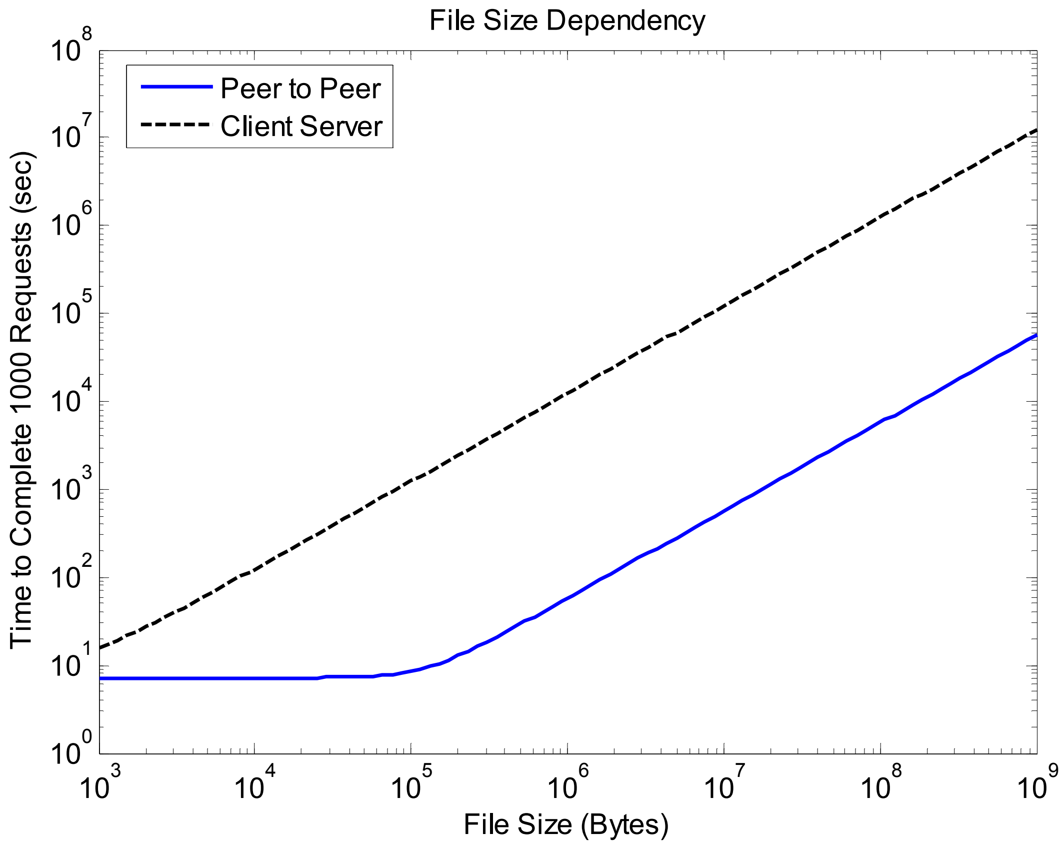

Figure 10 The dependency of transfer time on file size.

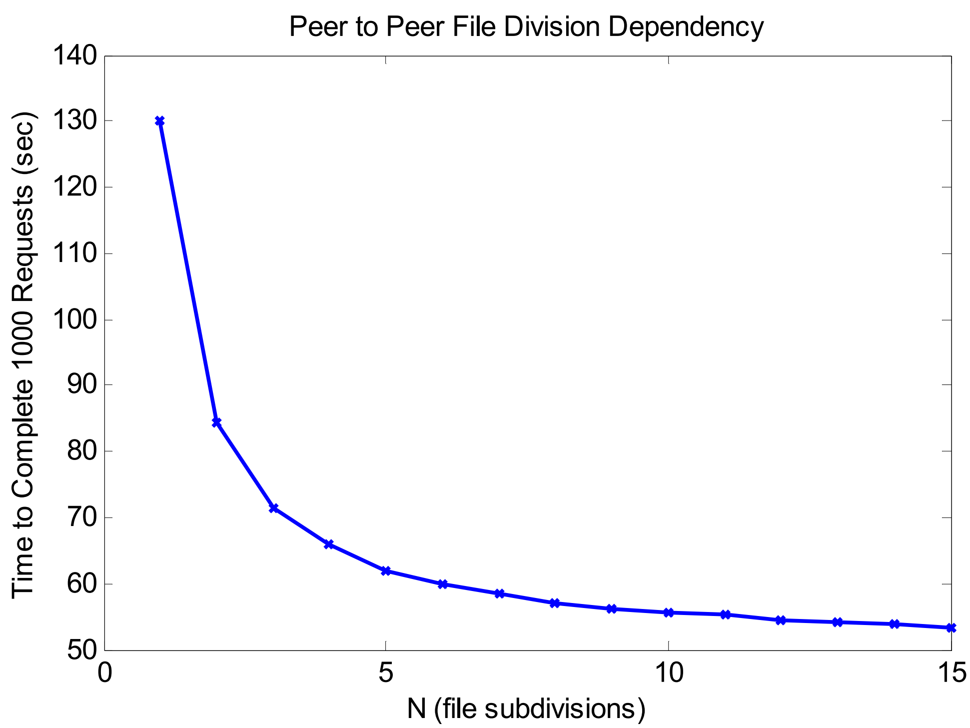

Figure 11 The dependency of transfer time on the number of file divisions.

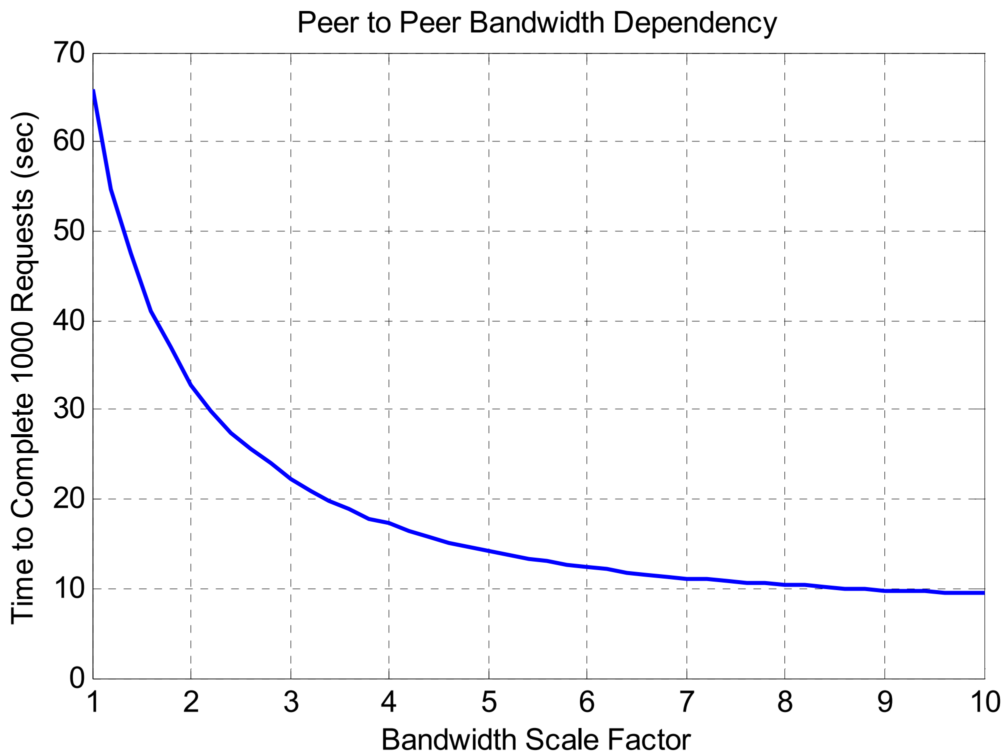

Figure 12 The dependency of transfer time on bandwidth for a peer to peer network.

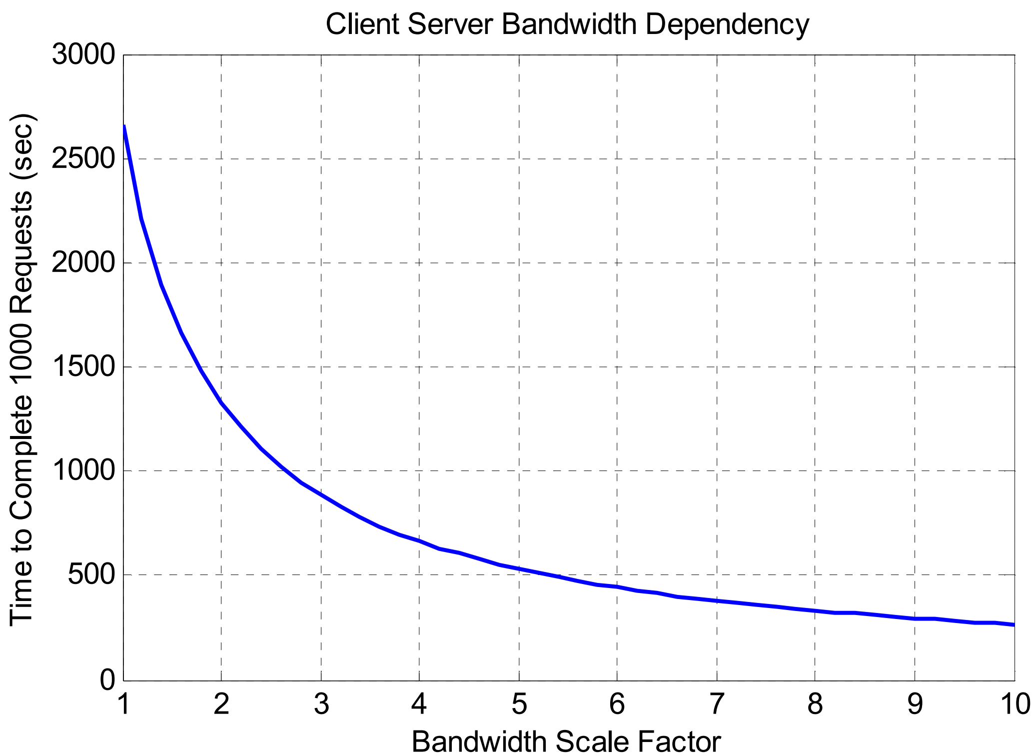

Figure 13 The dependency of transfer time on bandwidth for a client server network.

Scenario 2

A steady flow of requests is modeled by adjusting values for the file request arrival rate, λ. If the request rate is less than the uploading bandwidth of a server, it is reasonable to believe that a client server network will be able to service the requests without a queue forming. Likewise, if λ exceeds the uploading bandwidth, then a steady increase in the number of backed up requests will form. Indeed, the model shows this process, and indicates the point at which a peer to peer file sharing strategy first becomes advantageous: When the request rate exceeds a server's uploading bandwidth. At or below this rate, the system is constrained by the downloading bandwidths of the clients and increasing the number of uploaders will not alleviate the constraint or expedite the file transfer.

Figure 14 shows the progression of file owners when λ is 90% of the server's uploading bandwidth and the remaining parameters are the same as in the previous scenario, summarized in Table 3. It is clear that the performance under this condition is identical for both file sharing methods. In Figure 15, λ is increased slightly to 110% of the server's uploading bandwidth. The divergence between the two methods is now apparent, with a slight advantage observed for the peer to peer networks. As λ is increased, the performance advantage of peer to peer networks will also increase. This trend is shown in Figure 16, where λ is increased as a factor of the uploading bandwidth.

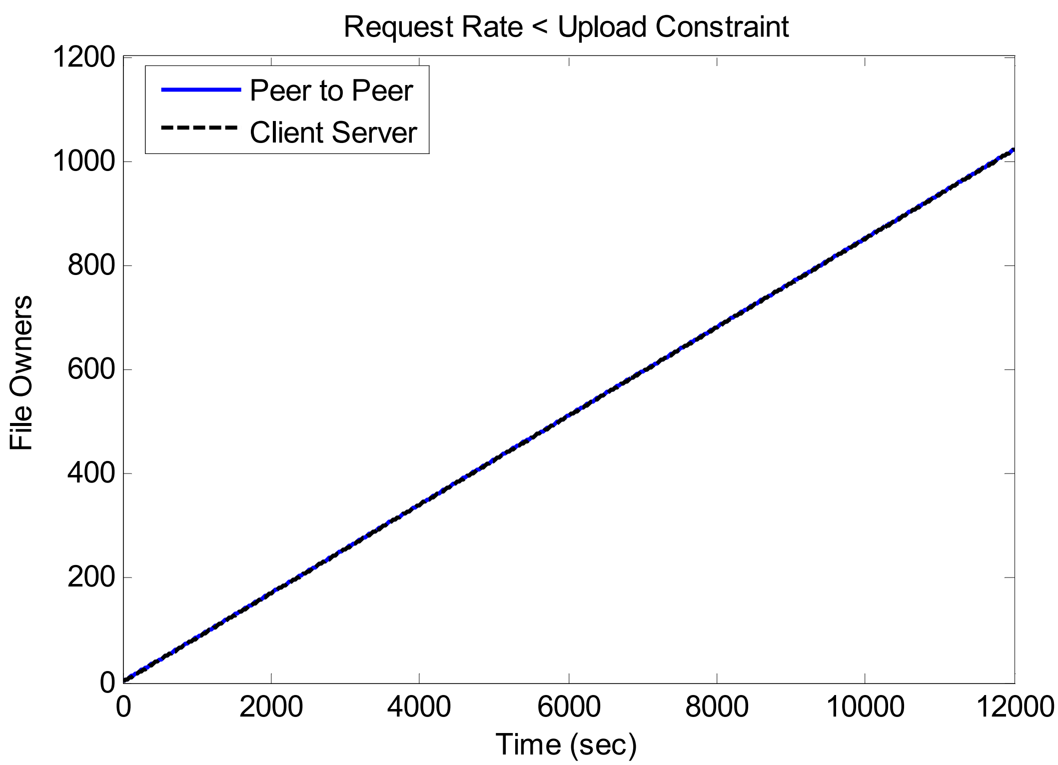

Figure 14 The number of completed downloads when the request rate for a file is less than the uploading bandwidth of a server.

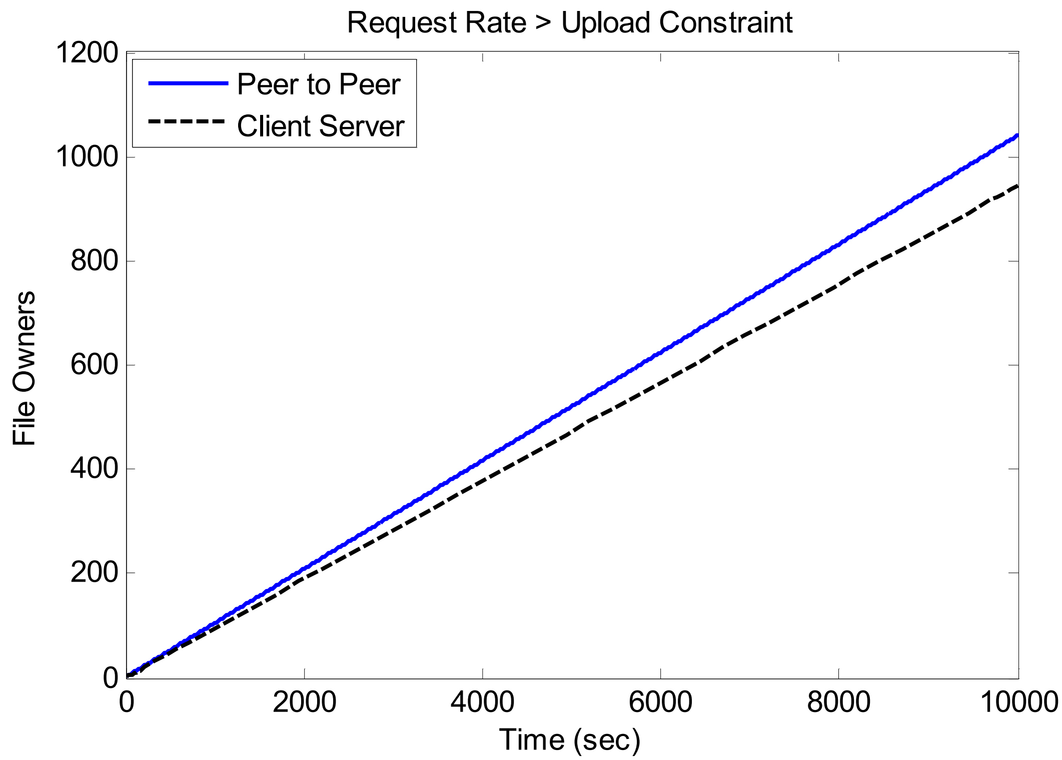

Figure 15 The number of completed downloads when the request rate for a file is greater than the uploading bandwidth of a server.

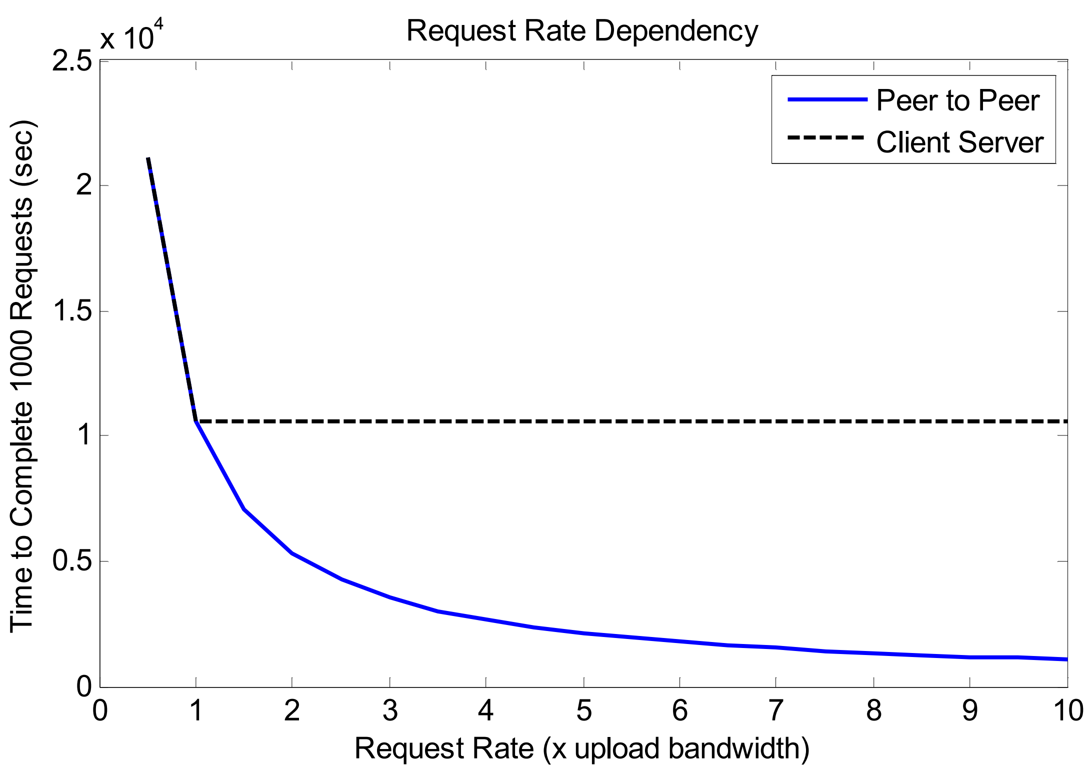

Figure 16 The dependency of transfer time on request rate.

Scenario 3

Scenario 3 considers transferring several files from a point of origin to a single destination. Conceptually, the models presented could be run for this scenario by simply running the implemented functions repeatedly until the number of files is transferred to a single owner. However, this approach is not appropriate for two reasons. First, the models do not represent this situation correctly. Because they are based on continuous functions, they do not correctly handle the discrete integral movement of a single peer from the downloading to the seed or owner category (continuous portions of a peer are transferred). Secondly, it is not necessary. The analysis is lucid if one simply considers the operation of the peer to peer network and realizes that gains are realized only after a downloading peer begins to upload to another peer. In this case there are no other peers, and each file will simply be transferred directly from the origination to the destination at a rate determined by either the uploading or downloading bandwidths.

Conclusions

Comparisons of the performance of peer to peer and client server networks are made for three different scenarios. These scenarios show when peer to peer networks begin to be advantageous, are extremely advantageous, or offer no advantage over client server networks. In this work, the time to complete a specific number of file transfers is the metric used to determine performance. When a network is upload constrained (i.e. a bottle neck exists on the uploading of files to the network), then a peer to peer network is advantageous because downloaders also upload to peers, causing an initial exponential increase in the uploading bandwidth. The greatest difference between the peer to peer and client server performance is observed when there is a sudden explosion of requests for a particular file. With the parameters used herein, the time required to fulfill the requests for the client server network is over two orders of magnitude greater than for the peer to peer network. When there is a steady flow of file requests, the client server network will perform at least as well as the peer to peer model if the rate of requests is equal to or less than its uploading bandwidth. In the case of transferring multiple files from a point of origin to a single destination, there is no advantage of using a peer to peer network over a client server one.

References

[1] X. Yang and G. d. Veciana, "Service Capacity of Peer to Peer Networks," presented at IEEE Infocon 2004.

[2] D. Qiu and R. Srikant, "Modeling and Performance Analysis of BitTorrent-Like Peer-to-Peer Networks," presented at SIGCOMM '04, Portland, Oregon, USA, 2004.

[3] K. K. Ramachandran and B. Sidkar, "An Analytic Framework for Modeling Peer to Peer Networks."