Quantifying Disincentives in Peer-to-Peer Networks

Michal Feldman† mfeldman@sims.berkeley.edu
Kevin Lai‡ laik@cs.berkeley.edu
John Chuang† chuang@sims.berkeley.edu
Ion Stoica‡ istoica@cs.berkeley.edu

†School of Information Management and Systems
U.C. Berkeley

‡Computer Science Division
U.C. Berkeley

1 Introduction

Cooperation is a central tenet of peer-to-peer systems. Without cooperation, users of a file-sharing system such as Gnutella suffer long download delays, if they are able to download at all [1]. Unfortunately, users have a natural disincentive to cooperate because allowing uploads may delay their own file downloads (see Section 2.1). In general, a disincentive to cooperate leads to the "tragedy of the commons" [6], where participants maximize their own utility at the expense of the overall utility of the system. However, few existing systems provide integrated incentives to counteract disincentives. Existing systems typically rely on altruism, which inevitably fails when participants cannot hold each other accountable for selfish behavior, or a user-managed incentive system, which overburdens users by requiring them to approve or reject every transaction.

In this paper, we use modeling and simulation to better understand the effects of cooperation on user performance and to quantify the performance-based disincentives in a peer-to-peer file sharing system. This is the first step towards building an incentive system. For the models developed in this paper, we have the following results:

- Although performance improves significantly when cooperation increases from low to moderate levels, the improvement diminishes thereafter. In particular, the mean delay to download a file when 5% of the nodes share files is 8x more than when 40% of the nodes share files, while the mean download delay when 40% of the nodes share is only 1.75x more than when 100% share.
- There is a high *potential* disincentive for sharing. Nodes with a 1.5Mb/s incoming and 128kb/s outgoing bandwidth (e.g., ADSL) can experience a 5x increase in the delay of their downloads when they allow uploads.
- In a homogeneous system, there is little *actual* disincentive to share, i.e., the difference in performance seen by a user that shares and a user that does not share is minimal. This is because in most cases the download latency is dominated by the sender's outgoing bandwidth rather than the impact of the receiver's uploads.
- In a system of nodes with heterogeneous bandwidths, there can be a high actual disincentive to share if a high percentage of either the low capacity or high capacity nodes share files.
- Prioritizing TCP acknowledgement packets (acks) over data packets eliminates the *potential* cost of sharing. However, while prioritization has a positive effect on the receiver's incoming throughput, it has a negative effect on the sender's outgoing throughput. As a result, the net effect of prioritization on system performance can be either negative or positive, depending on system parameters.

From these results, we draw the following conclusions:

- An incentive system should aim for the "sweet spot" in the sharing level because there is little benefit and possibly high cost (e.g., the CPU cycles, storage, and network bandwidth consumed by the incentive system) for increasing cooperation beyond that point.
- What determines if a user shares or not is the *perceived* sharing cost, which may be shaped by the *potential* cost more than by the *actual* cost.
- Prioritization of acks eliminates the *potential* cost, thus users' perception. We therefore expect prioritization to increase the sharing level of the system, which will consequently increase system performance.

2 P2P Performance Model

We develop a simple model to quantify a user's performance as a function of the percentage of users that share their resources in a peer-to-peer file sharing system. We use the average latency of a file transfer as the performance metric. More precisely, our goal is to capture how the performance experienced by a user varies as a function of (1) the sharing level, (2) whether a user shares files or not, (3) the asymmetry in the host incoming and outgoing bandwidths, and (4) the system load.

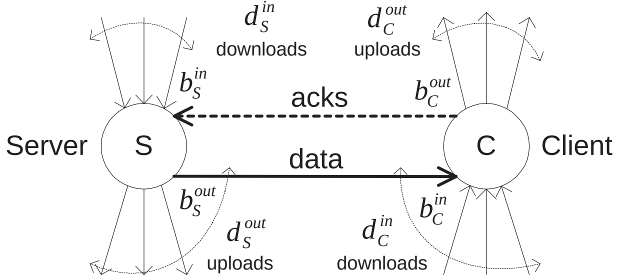

Figure 1: This figure shows a local view of one host downloading a file from another.

We make several simplifying assumptions: 1) searches are always successful, accurate, experience no delay, and consume negligible bandwidth, 2) files have the same size, popularity, and distribution, 3) a host downloads at most one file at a time, 4) and a single host's decision to share or not share has a negligible effect on the overall performance of the other users.

Figure 1 shows client C downloading a file from server S. The incoming and outgoing bandwidths of the server and the client are denoted by b_C^{in}, b_C^{out}, b_S^{in}, and b_S^{out} respectively. We assume that the bottleneck is always at the edge of the network, therefore we do not model the bandwidth of intermediate links between the client and server. Furthermore, we assume that the download time is dominated by the transfer time, and thus the propagation delay can be neglected.

Let d_C^{in}, d_C^{out}, d_S^{in}, and d_S^{out} be the number of incoming and outgoing data flows at the client and server, respectively. We first model user performance in a homogeneous system, and later extend the model to account for node heterogeneity in terms of bandwidth capacity and sharing level. In heterogeneous systems, we assume that hosts receive a number of download requests proportional to their capacity.

2.1 Effect of Uploads on Downloads

In this section we explain the reason for the *potential* disincentive for sharing. A user's basic rationale for disabling sharing is that his or her downloads will be delayed by uploads to other users. However, the cause and degree of this sharing penalty is not well understood. In particular, file download latencies may be influenced not just by the receiver's incoming bandwidth (b_C^{in}) and the sender outgoing bandwidth (b_S^{out}), but also by the receiver's outgoing bandwidth (b_C^{out}) and the sender's incoming bandwidth (b_S^{in}). This occurs because the TCP acknowledgement packets (acks) travelling from C back to S may compete against other flows for the b_C^{out} and b_S^{in} capacities.

The throughput of a TCP transfer depends on the interactions between the data and ack flows. Consider a sharing node x, which is downloading a file from node y and uploading a file to node z. Its outgoing acks to y may be competing with the outgoing data to z for the outgoing link capacity. As a result, the acks are delayed, causing an increased round trip time (RTT) between x and y, and a corresponding drop in download throughput.

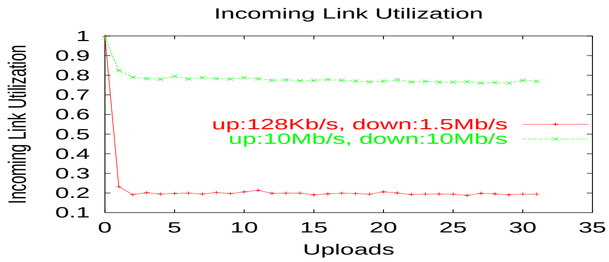

Figure 2: The utilization of the incoming link as a function of the number of outgoing flows

We verify this phenomenon using the `ns` simulator, and quantify the resulting incoming link utilization ρ for different classes of nodes. The topology consists of three nodes, x, y, and z, where x is downloading a file from y and uploading a variable number of files to z. The nodes are connected with FIFO tail-drop links. We use the FullTCP implementation with a segment size of 1460 bytes and one ack for every two segments. We repeat the simulation with different link bandwidths: 1.5Mb/s and 128kb/s for the incoming and outgoing links of DSL nodes, and 10Mb/s and 10Mb/s for the incoming and outgoing links of Ethernet nodes.

Figure 2 shows the utilization of the incoming link as a function of the number of outgoing flows. When the node is not uploading any files, the entire link capacity is utilized for its own download. However, as soon as it begins to upload a single file, the utilization drops to 0.8 for the Ethernet node and 0.2 for the DSL node. The utilization level does not drop further with additional outgoing flows. This behavior is caused by an increase in the measured RTT for the file download. The acks for the download must compete with the data packets of the uploads. The node's outgoing queue quickly fills up causing the acks to experience longer delays or drops. Since TCP throughput is inversely proportional to RTT and square root of the drop rate, the download throughput is reduced. Our simulations also show that downloads do not significantly affect uploading throughput.

2.2 Download Latency

2.2.1 Homogeneous system

Consider a system of n nodes, where $p_{sharing}$ is the percentage of nodes which choose to allow file uploads. Let ρ be the incoming link utilization as defined in the previous section. Thus the sharing nodes have $\rho < 1$ while the non-sharing nodes have $\rho = 1$. Let L be the average load (in file requests) placed on the system by each user. Assuming that the source and sink of the system load are uniformly distributed, then $d^{in} = L$ and $d^{out} = L/p_{sharing}$. For a file transfer from node S to node C, the available bandwidth may be limited by either the sender's outgoing or the receiver's incoming bandwidth. The available incoming bandwidth is b_S^{out}/d_S^{out} or $b_S^{out} * p_{sharing}/L$, and the available incoming bandwidth is $b_C^{in} * \rho/d_C^{in}$ or $b_C^{in} * \rho/L$. The effective transfer bandwidth will be the minimum of the two:

$$bw = min(\frac{b_S^{out} * p_{sharing}}{L}, \frac{b_C^{in} * \rho}{L})$$

Correspondingly, if it takes s_{file}/bw to transfer of a file of size s_{file}, the effective transfer latency is:

$$lat = max(\frac{s_{file} * L}{b_S^{out} * p_{sharing}}, \frac{s_{file} * L}{b_C^{in} * \rho})$$

We make several observations from this model. First, the bandwidth bottleneck switches between the sender side and the receiver side depending on the sharing level. In particular, the sender's outgoing link is the bottleneck when $p_{sharing} < \rho * \frac{b_C^{in}}{b_S^{out}}$, and the receiver's incoming link is the bottleneck when $p_{sharing} > \rho * \frac{b_C^{in}}{b_S^{out}}$. Intuitively, when few nodes share, those that do share have to perform a larger number of file uploads, and hence are more likely to become bottlenecks.

Second, when the receiver is the bottleneck, the latency is dependent on the incoming link utilization factor ρ. In this case, the decision to share ($\rho < 1$) or not share ($\rho = 1$) can affect download performance. If a node chooses to share, it will incur a slowdown penalty of $1/\rho$. This penalty can be significant for DSL nodes, where ρ can be as low as 0.2.

On the other hand, when the sender is the bottleneck, the download latency is inversely proportional to the sharing level $p_{sharing}$, but independent of ρ. In this case, the receiver will not incur any penalty for sharing.

2.2.2 *Heterogeneous system*

Consider a system that consists of two types of nodes, A and B, where f_x and $P_{sharing}^x$ represent their fraction in the system and sharing level, respectively. We assume that hosts receive a number of download requests proportional to their outgoing bandwidth. Therefore, the two types of nodes have the same *effective* outgoing transfer bandwidth. The expected effective outgoing bandwidth is: $(f_A P_{sharing}^A b_A^{out} + f_B P_{sharing}^B b_B^{out})/L$.

Sharing cost will be incurred only when the receiver incoming is the bottleneck. This happens when: $\frac{f_A P_{sharing}^A b_A^{out} + f_B P_{sharing}^B b_B^{out}}{L} > \frac{\rho_c * b_c^{in}}{L}$ (where b_c^{in} and ρ_c denote the incoming bandwidth and utilization factor of the client, respectively).

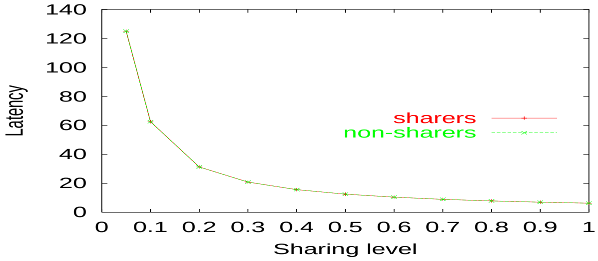

Figure 3: This figure shows the mean latency with respect to the sharing level in a homogeneous system of asymmetric (B-type) nodes.

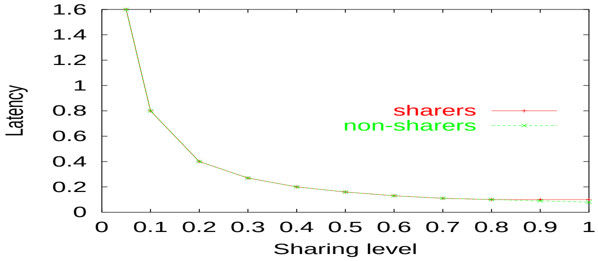

Figure 4: This figure shows the mean latency with respect to the sharing level in a homogeneous system of symmetric (A-type) nodes.

3 Results

In this section we present the results of our analytical model. We consider two types of nodes: A and B. Nodes of type A have $b^{in} = b^{out} = 10Mb/s$, and nodes of type B have $b^{in} = 1.5Mb/s$, and $b^{out} = 128Kb/s$.

3.1 Homogeneous System, Asymmetric Nodes

Figure 3 shows that in an homogeneous system consisting of nodes with asymmetric bandwidths, a sharing node experiences the same download latency as a non-sharing node. This is contrary to intuition and the micro-level simulation in Section 2.1 which shows a high potential cost for sharing (i.e., $\rho = 0.2$). However, the high bandwidth asymmetry causes the bottleneck to always be at the sender, thus negating the effect of the low ρ (i.e., $b_C^{in} \gg b_S^{out}$, such that $\rho * \frac{b_C^{in}}{b_S^{out}} > 1$). Consequently, there is no cost for sharing, regardless of the sharing level.

Figure 3 also shows that the relationship between sharing level and latency fits the "law of diminishing returns" [7]. A sender-side bottleneck causes latency to be proportional to $1/p_{sharing}$. As a result, increasing sharing from 5% to 40% reduces latency by a factor of eight, but increasing sharing from 40% to 70% only reduces latency by a factor of 1.75. 40% appears to be the *knee* beyond which increasing sharing provides little benefit. Figures 4 and 5 also show this effect. The diminishing returns are caused by the system's capacity

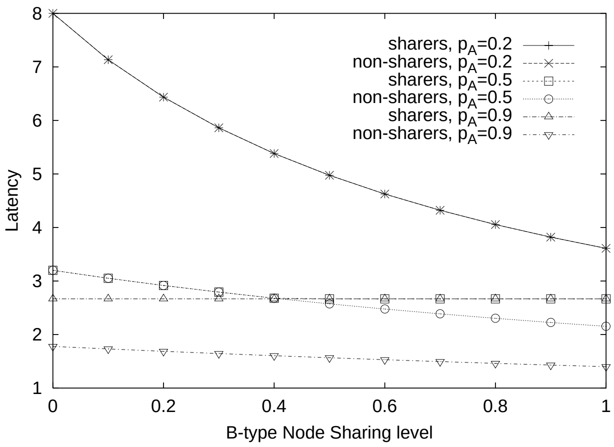

Figure 5: Avg latency vs. sharing level in a heterogeneous system with 95% B-type nodes and 5% A-type nodes

exceeding its load. At higher than 40% sharing, many nodes are serving at most one client, and so having more nodes share the same file does not help. This assumes clients cannot download different parts of the same file from multiple servers simultaneously and that higher levels of sharing do not result in a higher download rate. As a result of the diminishing returns effect, we conclude that an incentive system should aim for the "sweet spot" because there is little benefit and possibly high cost for increasing sharing beyond that point.

3.2 Homogeneous System, Symmetric Nodes

Figure 4 shows that in a homogeneous system of nodes with symmetric bandwidths, there is a small cost for sharing at high levels of sharing and no cost at other levels. This is because at low levels of sharing ($p_{sharing} < \rho$, from Section 2) , the bottleneck is at the sender and a sender-side bottleneck is not affected by the receiver's uploads. At high levels of sharing ($p_{sharing} > \rho$), the bottleneck is at the receiver, but since $\rho = 0.8$ for these nodes (from Section 2.1) and from the diminishing returns effect, the actual cost for sharing is minimal. This is evident in the small gap between the plots for sharing and non-sharing latency in Figure 4.

3.3 Heterogeneity

Recent studies [1, 9, 10, 8] show that peer-to-peer file sharing systems exhibit a large degree of heterogeneity in their bandwidths. In this section, we show that a heterogeneous system can have a high cost for sharing if many high bandwidth nodes share, but for a heterogeneous system with the current levels of sharing, there is little cost for sharing.

We consider a heterogeneous system that consists of 95% B-type nodes and 5% A-type nodes, where $p^B_{sharing}$ and $p^A_{sharing}$ denote their respective sharing levels. Load is distributed between the nodes in proportion to their outgoing bandwidths.

Substituting these values into the model presented in section 2, sharing cost is only incurred when: $500,000P^A_{sharing} + 121,600P^B_{sharing} > \rho_c b^{in}_c$. Notice that $\rho_A b^{in}_A = 0.8 * 10Mb/s = 8Mb/s$, therefore, A-type nodes never incur a penalty for sharing. However, B-type nodes may have a sharing cost if: $5P^A_{sharing} + 1.216P^B_{sharing} > 3$. From this equation it is clear that the higher $P^A_{sharing}$ is, the higher is the sharing cost incurred by B-type nodes.

Figure 5 shows the average latency for B-type nodes as a function of $p^B_{sharing}$ for three different levels of $p^A_{sharing}$. The cost of sharing depends on the degree of sharing in the high-speed nodes ($p^A_{sharing}$), as follows:

- For $p^A_{sharing} = 0.2$ (low): the bottleneck is at the sender. Therefore, sharing is costless regardless of $p^B_{sharing}$. The latencies for the sharers and non-sharers are shown by the two identical curves at the top of Figure 5.
- For $p^A_{sharing} = 0.5$ (medium): the bottleneck may be at the sender or the receiver, depending on $p^B_{sharing}$. There exists a crossover point (0.4 in this case) beyond which sharing becomes costly, and the cost increases with $p^B_{sharing}$. The sharers and non-sharers experience the same latency up to the 40% sharing level. Beyond 40%, only the non-sharers experience further reductions in latency.
- For $p^A_{sharing} = 0.9$ (high): the bottleneck is at the receiver. Therefore, sharing is costly regardless of $p^B_{sharing}$. While the non-sharers experience latency reductions with increasing sharing, the sharers' latency is limited by their incoming link utilization factor ρ. The latency curves for the sharers and non-sharers do not intersect.

In general, we observe that increasing sharing levels in the system leads to a shift of the file transfer bottleneck from the sender to the receiver. This leads to a difference in download latency between the sharers and the non-sharers. Hence, the cost of sharing increases with the level of sharing. However, there are many scenarios under which the sender is always the bottleneck, in which case sharing is costless. In fact, studies [1, 9, 10, 8] show that the levels of sharing of both low and high capacity nodes is very low. Consequently, there is little cost for sharing in such systems.

4 Prioritizing Acks

The results from the previous section show that a high sharing level increases system performance. To increase the sharing level we have to identify the source of the disincentive and try to reduce it or avoid it altogether. In our case, the source of the disincentive is the delayed or dropped acknowledgements that cause a higher download latency for sharers. We can eliminate sharing cost by prioritizing acks over data on the outgoing link. Employing such an approach would have opposing effects on the incoming and outgoing throughput. In the incoming direction, the node can utilize its entire incoming bandwidth, so $\rho = 1$. However, in the outgoing direction, the node prioritizes acks over data, therefore the throughput decreases. Given these two opposite effects, we are interested in deriving the net effect of prioritizing acks on the system performance. We first present its effect on a system with a fixed sharing level. However, since prioritizing acks reduces the disincentive to share, we expect the sharing level to increase, and we analytically derive the condition under which the net effect will be positive under all circumstances.

4.1 The Model

The throughput is determined by the minimum of the server's effective outgoing bandwidth and the client's effective incoming bandwidth. Denote: $bw_s = (b_s^{out} * P_{sharing})/L$ to be the server's outgoing bandwidth, and $bw_c = (b_c^{in})/L$ the client's incoming bandwidth. Using these terms, we can express the throughput with and without prioritization for the server and the client: $eb_s^{NP} = bw_s, eb_s^P = bw_s * \beta_s, eb_c^{NP} = bw_c * \rho_c, andeb_c^P = bw_c$, where $\beta, \rho \in (0, 1)$. The positive effect on the incoming direction is expressed by $\rho = 1$, and the negative effect on the outgoing direction is expressed by $\beta < 1$.

In what follows, we derive the net effect of prioritization on sharers (S) and non-sharers (NS) in three different possible states of the system:

1. The sender is the bottleneck for both S and NS. This happens when: $bw_s < \rho_c bw_c \Leftrightarrow P_{sharing} < \rho_c * \frac{b_c^{in}}{b_s^{out}}$.
2. The sender is the bottleneck for NS, but the receiver is the bottleneck for S. This happens when: $\rho_c bw_c < bw_s < bw_c \Leftrightarrow \rho_c * \frac{b_c^{in}}{b_s^{out}} < P_{sharing} < \frac{b_c^{in}}{b_s^{out}}$.
3. The receiver is the bottleneck for both S and NS. This happens when: $\rho_c bw_c < bw_c < bw_s \Leftrightarrow P_{sharing} > \frac{b_c^{in}}{b_s^{out}}$.

Tables 1 and 2 present the net effect of prioritization on sharers and non-sharers in the three states.

In conclusion, for a fixed sharing level, the net effect of prioritization is negative in many cases. However, prioritization reduces the performance-related disincentive for sharing. Therefore, we expect it to increase the sharing level. We denote the percentage of sharing before and after prioritization by $P_{sharing}^{NP}$ and $P_{sharing}^{P}$. Since the effective outgoing bandwidth is a function of the sharing level, it is also affected by prioritization, so we denote eb_s^{NP} and eb_s^P to be the throughput at the sender before and after prioritization, respectively. Observe that if $eb_s^{NP} < eb_s^P$, the system performance is always better with prioritization. This happens when: $P_{sharing}^{P} > \frac{P_{sharing}^{NP}}{\beta}$. For example, for $\beta = 0.9$, it is enough that $P_{sharing}^{P} = 1.11 * P_{sharing}^{NP}$ for prioritization to increase system performance.

Effective bandwidth to Sharers		
No Priority	Priority	effect of priority
bw_s	$\beta * bw_s$	worse
$\rho_c bw_c$	$\beta * bw_s$	better/worse
$\rho_c bw_c$	$min(\beta * bw_s, bw_c)$	better/worse

Table 1: This table shows the effect of prioritization on sharers.

Effective bandwidth to Non-sharers		
No Priority	Priority	effect of priority
bw_s	$\beta * bw_s$	worse
bw_s	$\beta * bw_s$	worse
bw_c	$min(\beta * bw_s, bw_c)$	same/worse

Table 2: This table shows the effect of prioritization on non-sharers.

4.2 Effect of Prioritization in Heterogeneous Systems

In this section we quantify the effect of prioritization on the heterogeneous system presented in section 3. NS simulations indicate that β is approximately 0.9. Referring to the three possible states presented above, we observe that A-type receivers will always be in state 1, namely: without prioritization the bottleneck will be at the sender's side. Therefore prioritization will decrease the performance of A-type receivers (sharers and non-sharers) under fixed sharing levels. B-type receivers, however, can be in one of the first two states. It will be in state 1 when: $5P_A + 1.216P_B < 3$, in which case the results will be like those of A-type nodes, or in state 2 when: $5P_A + 1.216P_B > 3$. Under this scenario, prioritization will decrease the performance of non-sharers. The performance of sharers will improve if: $4.5P_A + 1.09P_B > 3$, or decrease otherwise. The system will not be in state 3 under the settings presented here.

Figure 6 presents the average latency of B-type nodes under different sharing levels of A-type and B-type nodes. In the upper graph, where P_A is low and there is no cost for sharing, prioritization decreases performance for sharers and non-sharers. In the middle graph, for an intermediate P_A, there is a range where prioritization increases performance for sharers but decreases that of non-sharers, and for a high P_A, where there is always a cost for sharing, prioritization always increases the performance of sharers and decreases the performance of non-sharers.

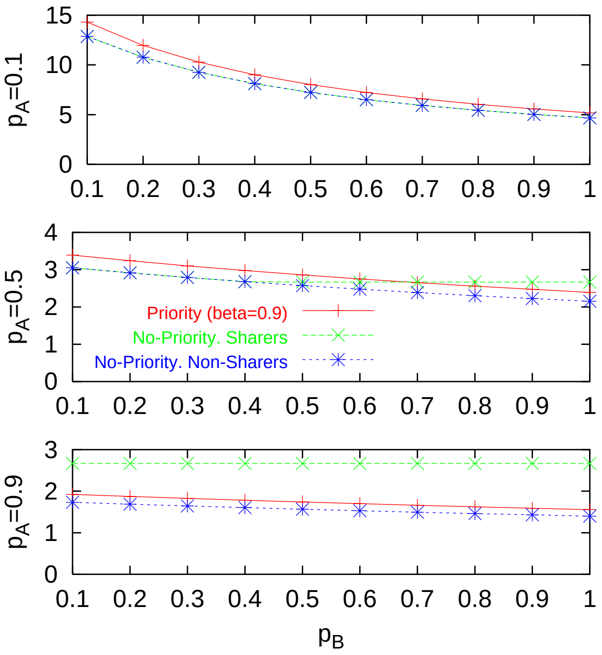

Figure 6: This figure shows the mean latency of B-type nodes in a heterogeneous system, under different sharing levels.

However, if the prioritization causes the sharing level to increase, the system performance will improve under all scenarios if: $5\mathrm{P}_{\mathrm{A}}^{NP} + 1.21P_B^{NP} < 4.5P_A^P + 1.09P_B^P$. Note also that in the case where A-type nodes and B-type nodes have the same sharing level ($P_B = P_A$), the condition under which prioritization always improves system performance is the same as in the homogeneous case, $P_{sharing}^P > \frac{P_{sharing}^{NP}}{\beta}$. For example, in figure 6, suppose that the initial state is a point in the upper graph. Even though prioritization increases average download latency for a fixed level of sharing, it is reasonable to predict that the sharing level will increase as a result of prioritization. This will take the system to a point in the lower graph, where the download latency is significantly lower.

5 Related Work

Several studies [1, 9, 10, 8] show that operational peer-to-peer networks have low levels of cooperation (e.g., 70% of hosts contribute no files).

Although previous work [4, 2] also models peer-to-peer file sharing networks, our work focuses on the difference in performance experienced by sharers and non-sharers rather than on the effect of non-sharers on the overall system. This difference between sharers and non-sharers is key because a performance-related incentive system must alter this difference to change the level of sharing in the system.

Some systems [3, 5, 11] implement or propose incentive mechanisms. However, as far as we know, there has not been previous work to understand how these systems change the performance difference between sharers and non-sharers.

6 Conclusion

This paper demonstrates that there is a high potential disincentive for sharing, but the actual disincentive may be either low or high, depending on the system characteristics. In situations where there is a high sharing cost or if users perceive it to be high, it will deter users from sharing their resources, thereby decrease the system performance. Since prioritization eliminates the potential cost, we expect it to increase the sharing level under these scenarios, thus increasing the system performance.

7 Acknowledgements

Antonio Garcia was one of the original developers of our simulator. Khian Hao Lim did extensive later development, including implementing a bandwidth model and doing automatic plotting of results.

References

[1] Adar, E., and Huberman, B. A. Free Riding on Gnutella. *First Monday 5*, 10 (October 2000).

[2] Asvanund, A., Clay, K., Krishnan, R., and Smith, M. Bigger May Not Be Better: An Empirical Analysis of Optimal Membership Rules in Peer-to-Peer Networks. In *Telecommunications Policy Research Conference* (2001).

[3] Clarke, I., Sandberg, O., Wiley, B., and Hong, T. W. Freenet: A Distributed Anonymous Information Storage and Retrieval System.

[4] Ge, Z., Figueiredo, D. R., Jaiswal, S., Kurose, J., and Towsley, D. Modeling Peer-to-Peer File Sharing Systems.

[5] Golle, P., Leyton-Brown, K., Mironov, I., and Lillibridge, M. Incentives For Sharing in Peer-to-Peer Networks. In *Proceedings of the 3rd ACM conference on Electronic Commerce, October 2001* (2001).

[6] Hardin, G. The Tragedy of the Commons. *Science 162* (1968), 1243–1248.

[7] Malthus, T. R. *An Essay on the Principle of Population*. 1798.

[8] Ripeanu, M., and Foster, I. Mapping the Gnutella Network: Macroscopic Properties of Large-Scale Peer-to-Peer Systems. In *Proceedings of the first Interntional Workshop on Peer-to-Peer Systemss* (2002).

[9] Saroiu, S., Gummadi, P. K., and Gribble, S. D. A Measurement Study of Peer-to-Peer File Sharing Systems. In *Proceedings of Multimedia Computing and Networking 2002 (MMCN '02)* (2002).

[10] Sen, S., and Wang, J. Analyzing Peer-to-Peer Traffic Across Large Networks. In *Proceedings of Internet measurement workshop 2002* (2002).

[11] Wilcox-O'Hearn, B. Experiences Deploying A Large-Scale Emergent Network. In *Electronic Proceedings if the International Workshop on Peer-to-Peer Systems* (2002).