

# Performance Evaluation of Proactive Peer-to-Peer File-Sharing Application for Releasing Network Congestion of Backbone Network in an Intranet

Chih-Chin Liang

Department of Business Administration, National Formosa University, Huwei Township, Chinese Taipei.  
Email: lgcwow@gmail.com

Received June 26<sup>th</sup>, 2012; revised July 24<sup>th</sup>, 2012; accepted August 3<sup>rd</sup>, 2012

## ABSTRACT

Peer-to-peer (P2P) application could be viewed as fast pull-based file-sharing system. P2P methods can deliver large files by dividing them into small chunks. However, P2P implementations employ greedy delivering strategies that can easily congest backbone network, but all kinds of data must be delivered on the backbone network. Through P2P methods, most of the connections available to a file host are occupied through pull-based methods by file retrievers. This limits the ability of the host to deliver any other data. P2P applications flood the backbone network with packets, thus leading to congestion. As a result, such P2P applications are banned on enterprise networks, where connections are expensive resources. Nevertheless P2P computing still retains significant advantages in file transmission. A delivering file can get to its destination through P2P application, but in the same time, the communication/delivery of other networking applications will be blocked. The data delivered/communicated through applications other than P2P one might have very important means for the management/business. Therefore, to utilize efficient P2P application on enterprise network is ideal, but the flooding of the backbone network by P2P chunks must be controlled. Thus, a P2P file-sharing application that actively manages network traffic would be ideal for the enterprise networks. Here, a proactive P2P (EP2P) file-sharing application proposed by Liang *et al.* (2009) that the performance has been proved by mathematic analysis and computer simulation could be considered as the solution that proactively manages network traffic. The best way to evaluate the system performance is through the real implementation on the network. In this study, the unit transmission time and block rate are evaluated as parameters determining the performance and cost of different file-sharing applications through 500 times of experiments. The experimental results show that through controllable P2P application, the manager could manage the bandwidth consumption of backbone network. The EP2P could be considered by the companies concerning on the balance between delivery efficiency and networking-traffic load.

**Keywords:** Network Traffic; Peer-to-Peer; Backbone Network; Network Management; Intranet

## 1. Introduction

Peer-to-peer (P2P) approaches are efficient pull-based mechanisms and are hence popular file-sharing tools [1]. BitTorrent, eMule, and Foxy are popular P2P applications [2,3]. In a P2P approach, a large file is divided into numerous chunks, each of which can be retrieved individually by the receiver through the connections available to the provider. After receiving chunks of the shared file, each file receiver then also acts as a file provider itself [3-6]. However, the host cannot control the traffic when using the abovementioned pull-based P2P methods, and hence there is a risk of heavy bandwidth loading. Once the retrievers engage all the networking bandwidth, the provider will be unable to retrieve content from other hosted applications. In other words, network bandwidth

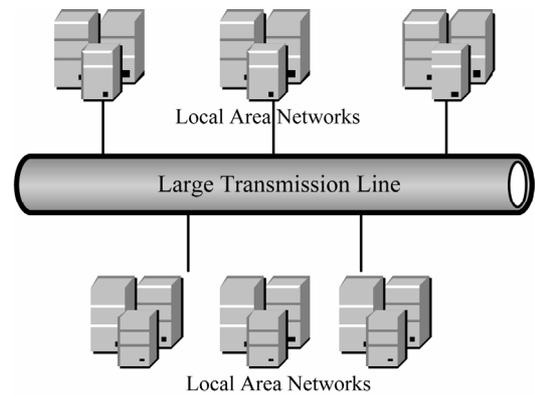
saturation makes it difficult for the provider to retrieve important files from other devices. Enterprise networks must ensure the retrieval of every kind of business-critical data packets [7,8]. Because connections are a limited resource, they must be optimized. Therefore, P2P applications that monopolize network bandwidth are impractical on enterprise networks.

Existing P2P applications address the abovementioned problems by a supply-side approach, mainly limiting the number of connections afforded by hosts to the P2P clients, leaving resources for other applications [9,10]. However, such solutions have some drawbacks. Connections can fall short in the face of a large number of peer requests. In addition, the backbone network is still vulnerable to congestion by a flood of P2P chunks from multiple clients. The backbone is a large transmission

line that carries data gathered from small local area networks that interconnect with it (**Figure 1**) [11,12]. For the company, the backbone network means the line that all networking devices must connect to.

The congestion on enterprise network also bothered Internet service providers (ISPs), because an ISP must provide the line that numerous customer's devices can connect to. In Taiwan, to reduce the backbone congestion, ISPs try to control backbone traffic on the demand side by limiting the user demands through control the bandwidth of user's computers. In the Taiwanese telecommunications oligopoly, the largest company, which owns Taiwan's backbone network, had attempted to solve the network traffic problems by limiting service use [13]. The company intended to launch a new Internet access service by the end of 2009 with downloads at 20 MBps (megabits packets per second) and uploads at 2 MBps *i.e.*, 20 MBps/2 MBps. However, one important aspect of the service was that once a subscriber transfers more than 200 GB (gigabytes) of data, that subscriber is penalized with a bandwidth limited to 10 MBps/2 MBps [14]. In other words, network resources are conserved by restricting services. However, with regards to consumer protection rights, such user-unfriendly services cannot be introduced in Taiwan. Therefore, the Taiwanese government ruled out the proposed service. However, the abovementioned incident reflects the key problem: the bandwidth of the backbone network is not always keeping up with the evolution of Internet services. The consumption of network bandwidth must be controlled.

Another demand-side solution is to curb greedy P2P routing behavior. Conventionally, P2P applications employ pull-based methods, which are the source of the congestion problem. However, the use of push-based file-sharing methods instead can allow hosts to actively manage traffic. Push-based P2P approaches can enable the host to equitably allocate bandwidth to each client because it is the sender that controls the file distribution [15-17]. In other words, push-based approaches could be utilized to manage the connections available to a host. A modified P2P approach named: EP2P utilized pull- and push-based methods has been proposed by Liang *et al.* (2009). The mathematical analysis and computer simulation have been adopted to evaluate the performance. However, the practical experiments have not been performed. The results of the mathematic analysis show that if the network is congested heavily, the transmission time through push-based method will be equal to pull-based method, and if the network not so congested, the transmission time though pull-based method will be smaller than push-based method. The simulation results show that the block rate (the packets that cannot be obtained by the receiver which is divided by the total packets that are



**Figure 1. Backbone network.**

delivered by the sender in a unit of time) will be rising along with the increasing number of file receivers [3]. Therefore, to find the performance and cost of EP2P, transmission time and network congestion must be found, because the increase in network congestion is a sensitive cost problem to manager and the decrease in transmission time is the major user's concern of file-sharing performance [3,8]. The cost and performance should be derived for user's consideration in the adoption of file-sharing applications.

The rest of this paper is organized as follows. Section 2 reviews related works on P2P approaches. Section 3 describes the EP2P infrastructure and EP2P application. Section 4 presents the results of a performance evaluation. Finally, Section 5 presents the conclusions.

## 2. Related Works

### 2.1. Pull-Based Mechanism

The P2P applications adopted pull-based methods to deliver files mainly. The pull-based method means that a receiver sends requests to the sender for retrieving an item, and then the required items are retrieved by the receiver actively [15-17]. The bandwidth of the sender is controlled by receivers. **Figure 2** shows the execution of pull-based mechanism.

Additionally, the correctness of the transferred packets under the pull-based mechanism is still needed to check. The approach to check the correctness of a retrieved file is checked by verifying the file size [17]. Once the file size is incorrect, the receiver retrieves the file again. After checking the correctness of the file, the procedure of file delivery has been finished.

Adopting a pull-based mechanism could make thousands of receivers retrieve files from a sender. However, such pull-based application has a defect: exhausting the connections of a sender when too many retrievers take back files simultaneously [15,17].

The modern P2P approaches can speed up the distribu-

tion of a large file among peers because the file is divided into chunks and each chunk is transmitted independently. All transmitted data will be verified through the exchange of the dynamic hash table (DHT) of file information in detail between peers [18]. A file is retrieved successfully after all chunks of the file are collected and merged correctly after verification [18,19]. The delivery load of the file provider could be shared to others. However, the efficient modern P2P approaches still exhaust the bandwidth of the backbone network, because of the large number of data transmission among peers.

Existing solution for relieving network congestion caused by a modern P2P method is restraining connections of the sender. However, the above solution is insufficient, because the uncontrollable requests for retrieving files from numerous retrievers. Too many requests also cause a congested backbone network that stops a device from retrieving needed information other than P2P packets. Fortunately, using push-based method, we might avoid congestion in backbone network [3].

### 2.2. Push-Based Mechanism

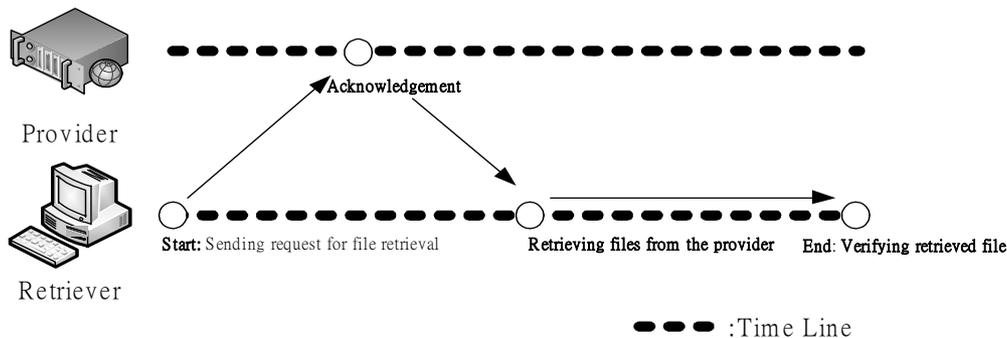
A push-based method represents a sender actively deliver files to receivers [7,8]. To adopt active method to transmit files, a sender could control its connections. However, through the theory of push-based mechanism, the sender controls no sent files. Therefore, the push-based method

must be fault-tolerant that ensures a packet is transmitted [4,20]. To ensure fault tolerance, two solutions are typically employed, namely: adding information to the sending files to verify the accuracy, and utilizing the signal to let senders and receivers decide the next action [3]. For example, whenever a sender sends a file to a receiver, the receiver could verify the correctness of the received part through comparing the real file size with the file size recorded in the information. A receiver could send a message to the sender to retransmit the file again whenever file size is incorrect. **Figure 3** shows the execution of the push-based mechanism.

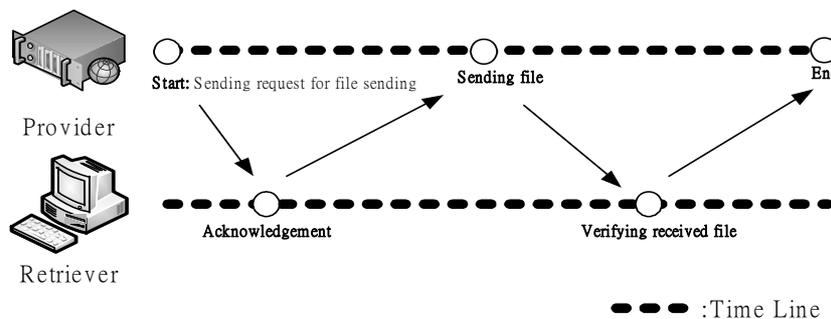
Actually, for file sharing on enterprise network, using push-based method or pull-based methods merely is inadequately. Through the pull-based method, the file delivery is fast, but the backbone traffic-load is heavy. Using the push-based method, the networking load is light, but the file delivery is inefficient. Therefore, it is might be a possible solution to build up a P2P application composed with pull- and push-based methods to deliver files with the balance between efficient file delivery and endurable backbone traffic-load.

### 3. System Implementation

Liang *et al.* (2009) have proposed an ideal model adopted pull- and push-based P2P mechanism to share files on enterprise network named: EP2P. The mathematical analysis and computer simulations have also been



**Figure 2. The process of pull-based mechanism.**



**Figure 3. The process of push-based mechanism.**

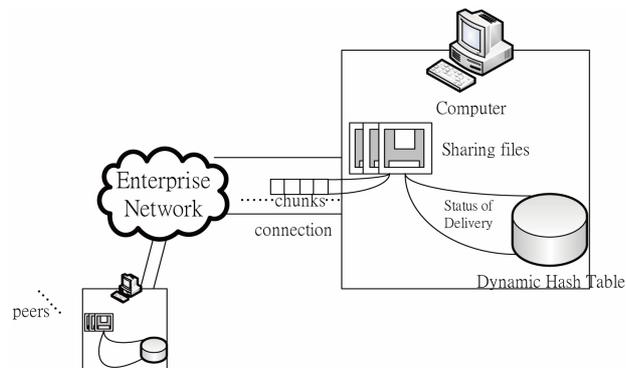
performed. However, the best way to evaluate the performance of file-distributing system is to implement it on a real network environment. The system infrastructure and evaluation methods to EP2P approach are introduced on following sections.

### 3.1. System Infrastructure

**Figure 4** shows the system infrastructure of EP2P. The EP2P application runs on a peer-to-peer environment. Each peer using EP2P approach has the same schema as other peers. A peer has a DHT used to store the status of the sharing file delivery. A sharing file is divided into chunks to be delivered. Using EP2P application, one peer can deliver chunks via push-based or pull-based method through connections. Because of the efficiency, the pull-based method is adopted to deliver chunks at first. Once the traffic is heavy, the push-based method should be used as the delivery approach.

Each peer uses DHT to record the status of sharing files. The recorded information includes *file names*, *owner names*, *file status in provider*, *completed parts*, *receiver names*, *transmission method*, and *file size*. Because a shared file using EP2P application must be divided into chunks for speeding up file delivery, the change of status of each chunk must be recorded to verify the correctness of transmission. Through comparing the size of received accumulating chunks with the information stored on DHT, the correctness of the transferring file should be assured. The *completed parts* shows the owned chunks of the sharing file. The *transmission method* is utilized for recording the delivery method. The *file size* is used for verifying the correctness of received chunks of the shared file. The above information recorded by a peer must be shared within peers and be updated whenever the file delivery through EP2P started or the status of the sharing file has changed.

Additionally, to reduce the backbone traffic, the changing policy of the delivery method must be conducted. Therefore, the opportunity to change the delivery



**Figure 4.** System infrastructure.

method from pull-based to push-based is important to decide based on congestion level. Therefore, in P2P design, three signals are devised, including: red, yellow, and green, to indicate the congestion level (Hsieh *et al.*, 2007). The congestion level is decided through comparing the original networking traffic with the current networking traffic. For example, the red signal means the traffic load is heavy; for example, we can set the red signal if the transmission speed is reduced to 60% of the original speed. The green signal indicates the networking traffic is light. In this work, the pull-based method can be utilized to transfer chunks while the signal is green and the push-based method can be utilized to transfer chunks while the signal is red. When the signal is yellow, the decision to utilize pull-based method or push-based method to deliver files depends on networking situations. If the networking speed is reducing gradually when the signal is green, and even the signal is yellow, the pull-based method is still adopted until the signal is red. If the networking speed is increasing gradually when the signal is red, the push-based method will be adopted until the signal is green.

### 3.2. Evaluation Method

This study uses cost and performance to help managers the adoption of P2P application. Because the transmission time of a file to all destinations (personal computers) can be viewed as the performance (The sooner the file can be destinations, the better file sharing is performed), transmission time could be used as the indicator of performance. The performance should be reflected to the unit time. The unit transmission time is computed as

$$T_u = \frac{\text{Transmission time}}{\text{The number of personal computer}}. \quad (1)$$

Additionally, the block rate could be adopted as the cost function on file delivery. Originally, the block rate has been defined as follow: “the packets that cannot be obtained by the receiver which is divided by the total packets that are delivered by the sender in a unit of time” (Liang *et al.*, 2009). However, the whole packets that cannot be obtained by the receivers are hard to find for the content delivery of P2P applications, because of a lot of communications and data exchanges among peers. Therefore, this work used optimal transmission speed and real transmission speed to represent the block rate (BR). The optimal transmission speed ( $S_o$ ) means the bandwidth of the network (the delivered packets per second). The real transmission speed means that the total delivered packets divided by total delivered time:

$$S_r = \frac{\text{Total Delivered Packets}}{\text{Total Delivered Time}} \quad (2)$$

The block rate should be as

$$BR = S_r/S_o \quad (3)$$

### 3.3. Implementation

The EP2P mechanism is initiated whenever a file needs to be transmitted. The status of the shared file must be necessarily recorded and updated. The host is identified as *sender name*; the owned chunks of the shared file, as *completed parts*; the name of the file, as file name; and the pull-based method, as *transmission method*. After the retriever duplicates the data transmitted by the host, the file is retrieved by the retriever.

Now, the pull-based procedure is initiated. The receiver retrieves chunks of the shared file from the host. Whenever the chunks are successfully delivered, the *completed parts* is updated, which takes place for all peers.

The receiver can retrieve missing chunks of the file, if any, from other peers who have these chunks. The pull-based procedure is complete once all the chunks of the file been successfully retrieved or the *transmission method* changes to “push-based method” when the signal turns red.

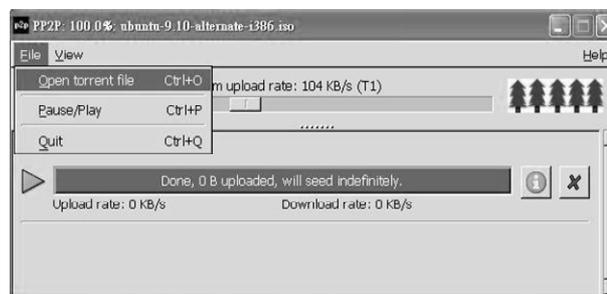
Initially, the pull-based method enables efficient file sharing. However, once network congestion sets in, we need the push-based method to ease the network load. The push-based method is used whenever the signal is red due to heavy traffic. The push-based method can enable the host to control the available connections. In addition, network bandwidth (congestion) can be controlled.

The push-based procedure is described as follows. The push-based procedure is initiated when the *transmission method* changes from “pull-based method” to “push-based method”. The sender actively sends chunks to each receiver who issues a request. Once all the chunks are successfully sent, the push-based method is completed. Once the signal turns green, the *transmission method* changes to “pull-based method”. In addition, if the receiver/sender is unable to communicate, the push-based method will be terminated.

## 4. Experimental Results

Actually, the best way to find the performance of software application is to implement it. To find the performance of EP2P, this work builds up the software product. The application is modified based on BitTorrent protocol mainly (**Figure 5**). Before sharing files, the seed (torrent file) must be built up.

Because the *transmission method* is assigned changeable along with the network congestion, the congestion must be defined. That is, using EP2P application, the



**Figure 5.** EP2P software application.

updated networking speed must be recorded on time. In this experiment, the speed is recorded whenever a set of chunks received/retrieved successfully. For example, once the networking speed is reduced from 300 KB (kilobits)/second to 100 KB/second, the signal is turn to red. The *transmission method* will be changed. Additionally, the sender delivers chunks to receivers through the push-based method are following round-robin fashion.

Additionally, the practical application can assign the different percentage of pull-based method and push-based method manually while sharing files. That is for the manager to control the congestion of backbone network. For example, if the network is congested, the network manager can change the delivery method from “automatic” to “manual” and set “the percentage of utilizing push-based method” to “70%” (30% of the file delivery will be performed by pull-based method during the unit time).

### 4.1. Environment

To understand the performance of EP2P application, this work builds up a lab for performance evaluation of following applications: eMule, BitTorrent, and EP2P. The experimental environment describes as follows. The experiment uses up to 50 personal computers in a lab, an all computers are with the same hardware and software. The CPU is Intel® Pentium IV with 3.00 GHz clock and the random access memory (RAM) are 504 MB (megabytes). The operation system is Microsoft® Windows XP with service pack 2. To simulate the congestion on lab network like on enterprise network, the networking bandwidth must be limited based on real case in this experimental environment and the file size must be large. In this work, the bandwidth is limited to 3 MBps. The large file with the size of 500 MB (megabytes) is created.

### 4.2. Results

After 500 times of experiments, this work found the experiment results (**Figures 6 and 7**). To find the comparison between P2P file-sharing systems, this work tests following P2P applications: eMule, BitTorrent, EP2P application,

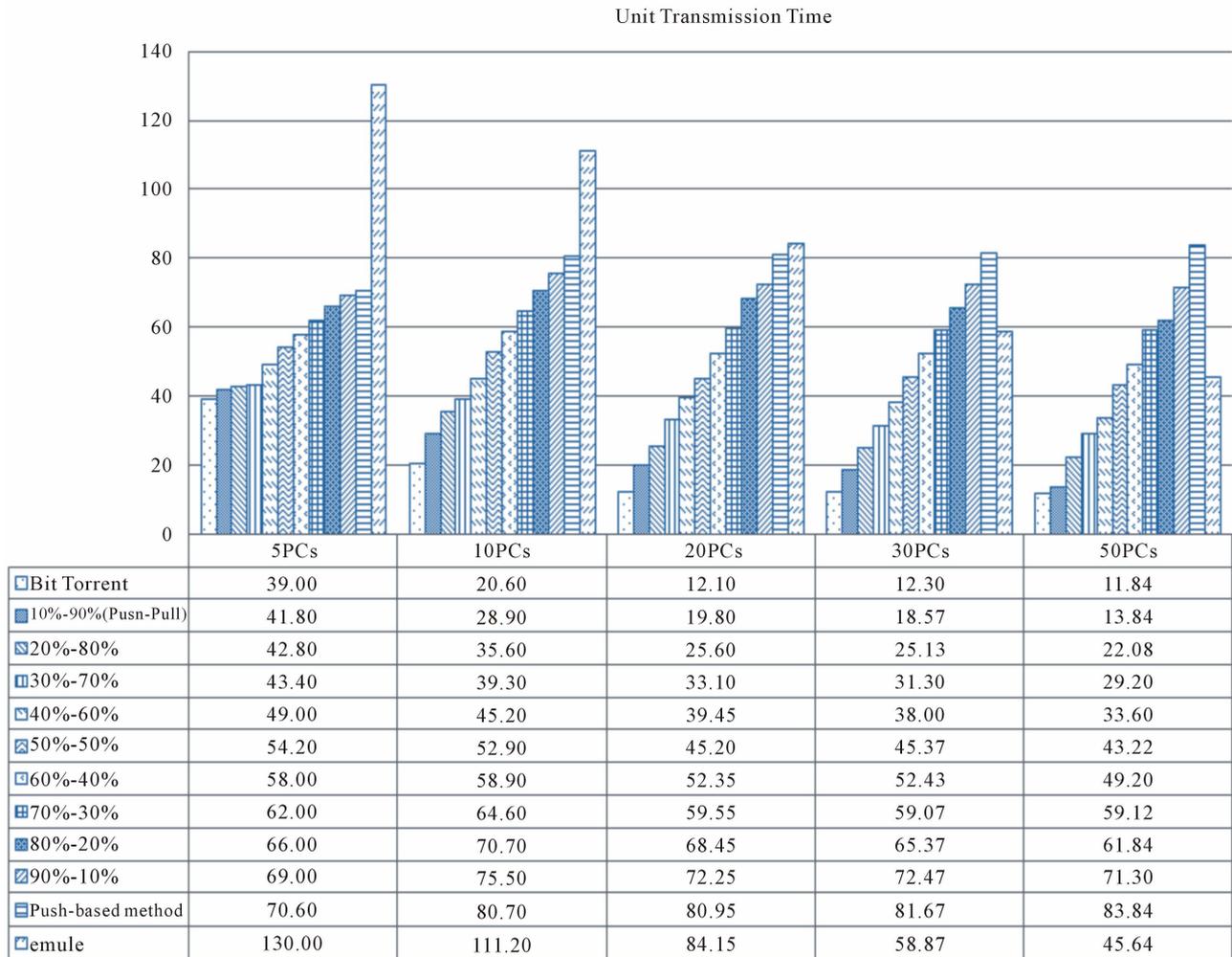


Figure 6. Unit transmission time (seconds).

and pure push-based method (modified based on EP2P application) through different number of personal computers (PCs). Additionally, to observe comparison results, for EP2P application, this work assigns pull- and push-based methods with different percentages manually during the transmission through EP2P.

Obviously, BitTorrent is with the best performance in file sharing, even the number of peer is large (Figure 6). Additionally, this work also found eMule is not a good pull-based P2P application. Through eMule, the unit transmission time ( $T_u$ ) is higher than through BitTorrent. Using eMule,  $T_u$  is even higher than through pure push-based method when the numbers of computer is <20 PCs. Additionally, using EP2P, the results of  $T_u$  are between using BitTorrent and pure push-based method.

The block rate ( $BR$ ) is conducted to help manager indicate the cost of file sharing (Figure 7). If  $BR$  is larger than one, the application causes congestion. The larger  $BR$  means the more congestion on network. The experi-

ment results show that BitTorrent is with the highest block rate. Additionally, the block rate of emule is small when the number of PCs is few (<30 PCs), but the performance (unit transmission time) of emule is poor. Through EP2P, the block rate is increase along with the increase in the percentage of the adoption of pull-based method to deliver files.

Therefore, through above experiment results, we find the EP2P should be considered to deliver files on the enterprise network, because we can deliver files with controllable backbone network. Through above two indicators: unit transmission time, and block rate, the manager could control the network congestion through EP2P approach. For example, if the congested network is anticipated, the manager could set EP2P application to manual mode, and set up the proper percentage of pull- and push-based methods based on situations. If the important files other than EP2P must be delivered on time on peak hour, the percentage of push-based method could

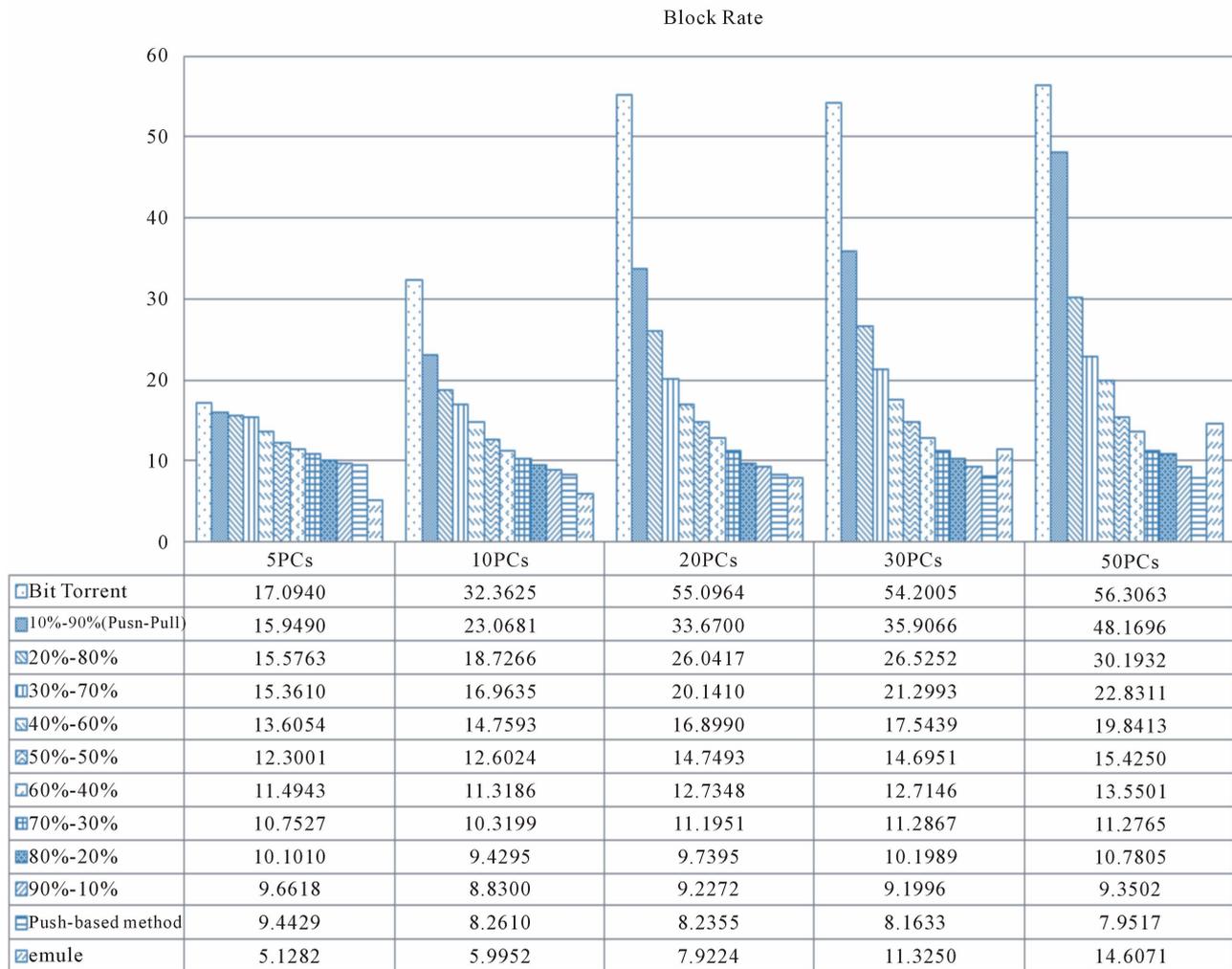


Figure 7. Block rate (times).

be set to a large value.

### 5. Conclusion

Networking resources are limited, and so congestion will delay or prevent the delivery of important data. Therefore, networking resources must be optimized, especially for enterprise networks. The modern pull-based P2P method is an efficient tool to deliver large files, but network congestion is the major drawback. Consequently, the P2P method is impractical in the cloud computing environment and on enterprise networks. Our study practices a hybrid P2P application, called EP2P (Liang *et al.* 2009), employing both pull- and push-based methods to deliver files. In addition, this study evaluated the performance and cost of different P2P methods via experiments. Two indicators are proposed for evaluating the performance and cost of file sharing: unit transmission time and block rate. The experimental results show that EP2P can help

balance transmission efficiency and network loads. The experimental results also show that once the network congestion sets in, the manager could use the push-based method to deliver the required files instead of pull-based method on time. Therefore, the EP2P application is practical to deliver files and can be used on the enterprise network.

### REFERENCES

- [1] N. Saxena, C. M. Pinotti and S. K. Das, "A Probabilistic Push-Pull Hybrid Scheduling Algorithm for Asymmetric Wireless Environment," *Proceedings of the IEEE GLOBALCOM*, Dallas, 29 November-3 December 2004.
- [2] L. Guo, S. Chen, Z. Xiao, E. Tan, X. Ding and X. Zhang, "A Performance Study of BitTorrent-Like Peer-to-Peer Systems," *IEEE Journal on Selected Areas in Communications*, Vol. 25, No. 1, 2007, pp. 155-169. [doi:10.1109/JSAC.2007.070116](https://doi.org/10.1109/JSAC.2007.070116)
- [3] C. C. Liang, C. H. Wang, H. Luh, P. Y. Hsu and W.Y.

- Yue, "Proactive Peer-to-Peer Traffic Control When Delivering Large Amounts of Content within a Large-Scale Organization," *Lecture Notes in Computer Science*, Vol. 5764, 2009, pp. 217-228.  
[doi:10.1007/978-3-642-04190-7\\_20](https://doi.org/10.1007/978-3-642-04190-7_20)
- [4] M. Bhide, P. Deolasee, A. Katkar, A. Panchbudhe, K. Ramamritham and P. Shenoy, "Adaptive Push-Pull: Disseminating Dynamic Web Data," *IEEE Transactions on Computer*, Vol. 51, No. 6, 2002, pp. 652-668.  
[doi:10.1109/TC.2002.1009150](https://doi.org/10.1109/TC.2002.1009150)
- [5] S. Herrería-Alonso, A. Suárez-González, M. Fernández-Veiga, R. F. R. Rubio and C. López-García, "Improving Aggregate Flow Control in Differentiated Services Networks," *Computer Networks*, Vol. 44, No. 4, 2004, pp. 499-512. [doi:10.1016/j.comnet.2003.12.007](https://doi.org/10.1016/j.comnet.2003.12.007)
- [6] X. Liu, J. Lan, P. Shenoy and K. Ramaritham, "Consistency Maintenance in Dynamic Peer-to-Peer Overlay Networks," *Computer Networks*, Vol. 50, No. 6, 2006, pp. 859-876. [doi:10.1016/j.comnet.2005.07.010](https://doi.org/10.1016/j.comnet.2005.07.010)
- [7] R. B. Dilmaghani and R. R. Rao, "Hybrid Wireless Mesh Network with Application to Emergency Scenarios," *Journal of Software*, Vol. 3, No. 2, 2008, pp. 52-60.
- [8] C. C. Liang, P. Y. Hsu, J. D. Leu and H. Luh, "An Effective Approach for Content Delivery in an Evolving Intranet Environment—A Case Study of the Largest Telecom Company in Taiwan," *Lecture Notes in Computer Science*, Vol. 3806, 2005, pp. 740-749.  
[doi:10.1007/11581062\\_81](https://doi.org/10.1007/11581062_81)
- [9] C. H. Wu, K. Chiang, R. J. Yu and S. D. Wang, "Locality and Resource Aware Peer-to-Peer Overlay Networks," *Journal of the Chinese Institute of Engineers*, Vol. 3, No. 7, 2008, pp. 1207-1217.  
[doi:10.1080/02533839.2008.9671475](https://doi.org/10.1080/02533839.2008.9671475)
- [10] M. Zhang, Y. Xiong, Q. Zhang and S. Yang, "Optimizing the Throughput of Data-Driven Peer-to-Peer Streaming," *IEEE Transactions on Parallel and Distributed System*, Vol. 20, No. 1, 2009, pp. 97-110.  
[doi:10.1109/TPDS.2008.69](https://doi.org/10.1109/TPDS.2008.69)
- [11] G. Kaur and D. Kumar, "MPLS Technology on IP Backbone Network," *International Journal of Computer Applications*, Vol. 5, No. 1, 2010, pp. 13-16.
- [12] R. M. Karp, C. Schindelhauer, S. Shenker and B. Vöcking, "Randomized Rumor Spreading," *Proceedings of the IEEE FOCS*, Redondo Beach, 12-14 November 2000, pp. 565-574.
- [13] F. Azzedin and S. Khwaja, "Towards Trustworthy Peer-to-Peer Environments: An Appraisal Analysis Approach," *Journal of Next Generation Information Technology*, Vol. 1, No. 1, 2010, pp. 61-76. [doi:10.4156/jnit.voll.issue1.6](https://doi.org/10.4156/jnit.voll.issue1.6)
- [14] W. Su, "Chunghwa Telecom Is Intended to Follow NCC Decision Temporarily to Promote 20 M Fiber Networking Service," 2010.  
<http://www.ithome.com.tw/itadm/article.php?c=57497>
- [15] H. C. Hsieh, C. J. Lee, P. C. Liu and W. K. Shih, "Evidence-Based Fault Detecting Scheme under P2P Streaming Network," *ICL Technical Journal*, Vol. 121, 2007, pp. 110-115.
- [16] D. Kopec and S. Tamang, "Failures in Complex Systems: Case Studies, Causes, and Possible Remedies," *Proceedings of the ACM SIGCSE Bulletin*, Vol. 39, No. 2, 2007, pp. 180-184. [doi:10.1145/1272848.1272905](https://doi.org/10.1145/1272848.1272905)
- [17] R. H. Wind, K. A. Griggs and E. Y. Li, "An Architecture for Distributed Scenario Building and Evaluation," *Communication of the ACM*, Vol. 48, No. 11, 2005, pp. 80-86.  
[doi:10.1145/1096000.1096009](https://doi.org/10.1145/1096000.1096009)
- [18] K. C. Tsai and C. Chen, "Effect of Heterogeneity on Static Load Balance Algorithm Performance in DHT Systems," *Journal of the Chinese Institute of Engineers*, Vol. 31, No. 2, 2008, pp. 301-311.  
[doi:10.1080/02533839.2008.9671383](https://doi.org/10.1080/02533839.2008.9671383)
- [19] S. Rieche, K. Wehrle, O. Landsiedel, S. Götz and L. Petrak, "Reliability of Data in Structured Peer-to-Peer Systems," *Proceedings of the HOT-P2P*, Volendam, 8 October 2004.
- [20] R. Lemos, P. A. C. Guerra and C. M. F. Rubira, "A Fault-Tolerant Architectural Approach for Dependable System," *IEEE Software*, Vol. 23, No. 2, 2006, pp. 80-87.  
[doi:10.1109/MS.2006.35](https://doi.org/10.1109/MS.2006.35)