Transfer Size Limitations: Types, Influencing Factors, and Possible Solutions

Prashanth Kodurupati

Information Technology, Managed File Transfer Engineer, PragmaEdge LLC, Alpharetta, United States of America
Email: prashanth.bachi21@gmail.com

Abstract File size is one of the reasons a transfer may fail between a client and host server. The primary reason for this failure is that at least one of the stakeholders in a transfer is unable to handle the large file size. This may reflect an MFT, recipient, sender, email agent, or cloud storage's limitations. Most of these are inherent, so the solutions tend to be external in nature. Compression is the most commonly employed solution to reduce the size of the file being transferred. However, it can also be split into smaller pieces and recombined after delivery or sent through a cloud storage intermediary.

1. Introduction

A file transfer taking place between two entities (B2B, B2C, or C2C) is bound by limitations imposed/inherent in the entities themselves (the systems they are using) and the transfer method/technology/tools chosen for the job. In every instance, the transfer has to comply with the lowest set of requirements. If company A wants to transfer a file to company B through a cloud mediator called XYZ, the transfers may be limited to 5 GBs if that's what company B is equipped to receive, even if both company A and the cloud can handle file transfers in terabytes. So in any transfer, the element with the lowest file size handling capabilities sets the benchmark.

2. Literature Review

Size has been a critical dimension when it comes to file transfer since the early days of computers and other electronic devices (like phones). Different transfer methods, file systems, devices, etc., have their own file size transfer (and, in some cases, storage) limitations. Over the years, several strategies have been adopted to get around these limitations, but while many of them work effectively against total data transfer volumes (several files of small sizes), relatively few solutions are available for transferring files that are above a certain size threshold [1].

The file size limitations may have been affected by the transfer protocols in the early days, but nowadays, the most commonly used set of protocols, even the secure ones like Secure File Transfer Protocol (SFTP), don't have any inherent file size limitations [2]. The same may be true for certain Managed File Transfer (MFT) systems/software, i.e., no file size limits when they are facilitating a transfer between two entities [3]. However, even if an MFT is in play but it's integrated with an email server or a file is being sent from an MFT system of company A to company B via email, the file size limitation of the email server will supersede [4].

3. Problem Statement: Large File Transfer Failures

The overarching problem that we are looking at in this article is a file transfer failing because of the file size. However, it may arise for a number of reasons.

3.1 MFT Limitations

Most of the commonly used MFT systems or MFT functionalities of business integration systems like IBM Sterling have little to no file size limitations. They may allow for files of several gigabytes to be transferred seamlessly. However, even with an MFT facilitating a transfer between two entities, there may be scenarios where the file transfer fails primarily due to the file size.

This includes how unattended transfers/unsupervised transfers are set up and their rules [5]. File size is one of the basic settings or rules in file transfers set up in an MFT, and the default settings may have a predefined file size. Any transfers exceeding that number may fail.

An MFT mismatch can also be an issue. If one entity is using an MFT while the other isn't and they are equipped to receive files in a conventional manner, i.e., an email agent or cloud link, the transfers would be limited by those tools. If both sender and recipient are using two different MFTs, the lowest transfer settings will define the size limit and may prevent a file of a larger size from transferring.

3.2 Sender's Limitations

An MFT may govern the transfer itself, but the size limitations may kick in before that. They may not have the right file system to create, store, copy, or internally transfer a file over a specific size limit. The cybersecurity rules, resource bandwidth, resource limitations, and several other settings or limitations may also influence an entity's ability to send a file of or beyond a certain size.

3.3 Recipient's Limitations

Similar to a sender's limitations, a recipient's limitations may prevent them from accepting a file transfer if the file size exceeds certain thresholds. This may include available space on the server/devices, file system, security considerations, transfer bandwidths, etc. These limitations go beyond the limitations inherent in the MFT system.

3.4 Email Agent's Limitations

When a file transfer is being made through an email agent, their file size limitations can prevent the transfer from happening. Different public email agents/email providers have their own file size limitations. Gmail only allows an attachment of 25 MB, and anything above that can be transferred through Google's own cloud. Each email provider/email agent may have its own size limitations.

The email functionality of systems like IBM Sterling that are used to integrate different businesses together may have different limitations. Some (like the MFTs) may not have any limitations, but even if they can send out a large file, that doesn't mean the recipient's email agent would be able to accept it.

3.5 Cloud Storage's Limitations

If a cloud is used as an intermediary for a transfer, i.e., the sender uploads a file on the cloud and shares the link with the recipient that they can use to download the file, the cloud's limitations may prevent the transfer of files beyond a certain size threshold. This is often governed by the user levels/tiers, with free or even some paid users only allowed single files under a predefined size limit. How much space the sender or recipient has left on their cloud storage is another factor that can influence the success or failure of a large file transfer.

3.6 Transit and Legacy Systems Limitations

In limited scenarios, the large file being transferred may be stored at a temporary third-party location for a limited time. Here, the limitations of that temporary storage may also come into play. If it can't handle a file beyond a certain size threshold, the transfer may fail.

4. Proposed and Implemented Solutions

A few solutions can be actively or passively implemented to get around the file size problem. The active solutions are implemented on a case-by-case basis and may or may not be automated. In contrast, the passive solutions might be inherent to the intermediaries of a file transfer, like an MFT, cloud, or email agent.

4.1 Compression

One of the most commonly implemented solutions to get around the file size limitations is compression. A large file can be compressed to a smaller version and then decompressed on the client side using the same protocol/methodology used to originally compress the file to ensure data integrity. How much data can be compressed can depend upon several factors, including data type, compression algorithm being used, hardware resources available for compression and decompression (with client), etc. In many cases, 10:1 data compression ratios, where a 10 GB file can be compressed into a 1 GB file, are quite common.

So, if you are limited to 5 GB file transfers and you have a 6 GB file that you have to transfer, you can lower its size to 5 GB through compression. Theoretically, you *can* lower the size even further, but that may not be ideal. Less compression results in more efficient use of computing resources and might allow for faster decompression at the client end. So, the typical goal should be to compress just under the size threshold.

It's important to note that there are two main types of compression - lossless and lossy [6]. As the name suggests, lossless compression ensures that no data is lost during the compression, while in lossy compression, the size may be reduced by getting rid of *some* data. The latter is common in image transfers.

4.2 File Splitting and Recombination

A file can be split into smaller components, and each component can be transferred as a separate entity, allowing you to overcome the limitations associated with transferring a single file that's too large. Once the transfer is complete, the components can be recombined to make the whole file again. The solution may work faster and more efficiently in some cases, but it has certain vulnerabilities and limitations. Both the sender and recipient should be in agreement when it comes to the splitting and recombination protocols/methodologies; otherwise, data integrity might be compromised. Similarly, even if a small chunk of the split file is corrupted or is unable to transfer due to networking issues, it may impact the entire file at the recombination stage. These limitations are among the reasons it's not an inherent solution in MFTs or email agents (among other transfer facilitators).

4.3 Cloud Storage Services

Another commonly implemented solution is using third-party cloud storage services that are available in a wide range of pricing, security, and performance tiers. The solution is rarely implemented when both client and host servers are using a similar MFT or business integration system like IBM Sterling. However, if the file has to be transferred via email or another channel because of an MFT mismatch or some other limitations, cloud storage can serve as a valid intermediary. The sender can upload the file to the cloud and send the recipient a link to the file. The recipient can download the file using that link. Additional security measures can be implemented, such as a specific email requirement to download the file or download credentials that can be sent to the recipient via a different channel if the download link email is compromised. File size limits vary among cloud providers and can easily be up to several GBs.

5. Use Cases

Problem	Solution	Use Case for Financial Sector
MFT Limitations	* MFT with Large File Transfer Support * Splitting/Chunking with Aggregation (if supported by MFT)	* Investment Bank: Transferring large volumes of market data or trade confirmations between the investment bank and its institutional clients. * Insurance Company: Transferring high-resolution medical images or large claim files between the insurance company and healthcare providers.
Sender's Limitations	* Compression	* Retail Bank: Sending compressed monthly account statements or tax documents to customers with limited email storage space.

Recipient's Limitations	* Cloud Storage Services * Secure File Transfer Protocol (SFTP)	* FinTech Startup: Sharing large software updates or data sets with early adopters who might have limited IT infrastructure. * Hedge Fund: Securely transfer confidential financial models or research reports to external analysts using SFTP.
Email Agent's Limitations	* Cloud Storage Services	* Financial Advisor: Sharing investment proposals or market research reports with clients who primarily rely on email communication.
Transit and Legacy Systems Limitations	* Secure File Transfer Protocol (SFTP)	* Central Bank: Securely exchanging regulatory reports or financial data with legacy systems of commercial banks. * Payment Processor: Transferring large transaction logs or batch files to/from legacy systems of partner merchants.

6. Conclusion

Transfer size limitations may stem from any "stakeholder" in the transfer, ranging from sender/recipient to MFTs or email agents. The right solution may depend upon the needs of the client and sender, the resources available, file size, data integrity and security concerns, and a number of other reasons. Security concerns are paramount for businesses in the financial, defense, and healthcare sectors. In contrast, many B2C businesses may focus more on ease of transfer. Understanding the root of the transfer size limitation is critical to choosing the right solution.

References

[1]. E. Yildirim, E. Arslan, J. Kim and T. Kosar, "Application-Level Optimization of Big Data Transfers through Pipelining, Parallelism and Concurrency," IEEE Transactions on Cloud Computing, pp. 63 - 75, 2015.

[2]. C. M. Nawej and P. A. Owolawi, "Evaluation and Modelling of Secured Protocols' Spent Transmission Time," in 2018 International Conference on Intelligent and Innovative Computing Applications (ICONIC), 2018.

[3]. O. T. S. MFT, "Solving the problem of sending ridiculously large files," Open Text, 2015.

[4]. C. P. Support, "Regulatory Compliance/HIPAA Security Rule Technical Safeguards," Journal of the California Dental Association, 2016.

[5]. IBM, "IBM Sterling Connect: Direct 5.3 Overview," IBM Documentation, 2016.

[6]. N. Sasaki, K. Sato, T. Endo, and S. Matsuoka, "Exploration of Lossy Compression for Application-Level Checkpoint/Restart," in 2015 IEEE International Parallel and Distributed Processing Symposium, 2015.