

# Cloud Based Managed File Transfer Services

Pranav Kulkarni, Sahil Jain, Vinay Kumar

*Department of computer Engineering,  
Sinhgad college of engineering ,Vadgaon bk.,  
Pune, India*

**Abstract-**In today's Era there is a need for Managed file transfer services over trivial one's mainly due to the Outsourcing done in Product Lifecycle Management and a need to get files from source to the destination (through a chain of people ) in a faster, more reliable way . By making use of these Managed File Transfer Services we are able to minimize the risk of unauthorized access and also able to keep a track of the files sent by/received to specify parties.

This product is provided as a SaaS (software as a service) to the clients, mainly for exchange of Big Data between related parties. The File Transfer is done using UDT (UDP based Transfer Protocol). The Product is not using the conventional FTP for the file transfer. The product can keep the information updated all the time so that the members get the updated information all the time.

With this we can Transfer the right files to the right place at the right time in the right format – and ensure information is secure at every step along the way. The Managed File Transfer (MFT) Services quickly and cost-effectively delivers large files to anyone, anywhere – regardless of platform.

## I. INTRODUCTION

Today, national or international high-speed networks have connected most developed regions in the world with fiber. Data can be moved at up to 10 Gb/s among these networks and often at a higher speed inside the networks themselves. For example, in the United States, there are national multi-10 Gb/s networks, such as National Lambda Rail, Internet2/Abilene, Teragrid, ESNet, etc. They can connect to many international networks such as CA\*Net 4 of Canada, SurfNet of the Netherlands, and JGN2 of Japan.

In today's Era the bandwidth of 10GB is no longer said to be a huge bandwidth as it is already in use in various countries. The effective use of this bandwidth is necessary in order to achieve very high transfer rates. TCP and other trivial file transfer mechanisms are inefficient to make use of the bandwidth. So there is a need to transfer the data in managed way. The cloud based managed file transfer uses UDT protocol; i.e. UDP based transfer protocol for transferring files at higher rates and efficient use of large bandwidth.

Many business projects are outsourced nowadays, which creates a need of file transfer mechanism for huge file with very high transfer rate with proper use of bandwidth. As the data is large there is possibility of network failure or any other network problems during the transfer of the data. So such broken downloads should not start from the beginning leading to larger time required. MFT will provide resume facility for such broken downloads. Also Cloud Based MFT will provide facilities of sending email notifications and account management.

## II. LITERATURE SURVEY

The FTP do not allow the encryption of data which lead to the attacks like packet sniffing, brute force attack. The FTP(File Transfer Protocol) which is based on TCP differs from the UDT in the way of utilization of bandwidth. The main problem with TCP is TCP demonstrates poor performance in long links especially when the bandwidth is high. The UDT utilizes 90% of the bandwidth with the help of new congestion control algorithm and bandwidth utilization technique. TCP's congestion control algorithm needs a very long time to probe the bandwidth. UDT employs an AIMD rate control algorithm that uses a bandwidth estimation technique to determine the best increase parameter for efficiency.

HTTP is insecure and is subject to man-in-the-middle and eavesdropping attacks, which can let attackers gain access to website accounts and sensitive information. HTTPS is designed to withstand such attacks and is considered secure against such attacks (with the exception of older deprecated versions of SSL). HTTPS can be much slower than HTTP. Depending on the strength of the cryptography, performance differences can become noticeable. Everything in the HTTPS message is encrypted, including the headers, and the request/response load.

UDP (User Datagram Protocol) is unreliable, connectionless protocol. UDT is built on top of User Datagram Protocol (UDP) by adding congestion control and reliability control mechanisms. UDT is an application level, connection oriented, duplex protocol that supports both reliable data streaming and partial reliable messaging.

Real-time streaming protocol uses a combination of protocols such as TCP (connection based protocol), UDP (connectionless protocol), and RTP to achieve various functions by maintaining session/state between server and client through an identifier. Streaming with RTSP fails if a firewall separates the client and server, and the firewall blocks the ports and protocols that RTSP uses. It also has the overhead of requiring multiple requests before playback can begin. If TCP is used for transmission of data then we have to deal with the overhead of retransmission of data and if UDP is used then overhead of packet loss. UDT uses UDP with reliability such that all the problems with TCP and UDP can be solved.

## III. IMPLEMENTATION

The cloud based MFT is created by making use of open source UDT services. UDT the UDP based transfer protocol

uses the bandwidth effectively and transfer the data at higher rate. The user has to register and create an account in order to make use of MFT services. The user can transfer the data to any third party with a valid Id. The files can be transferred to other cloud as well.

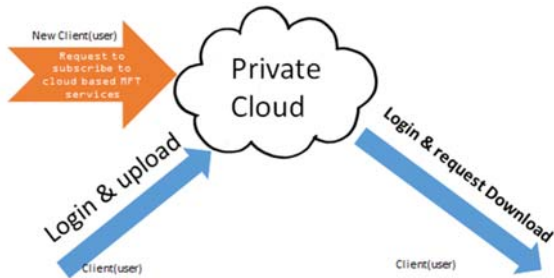
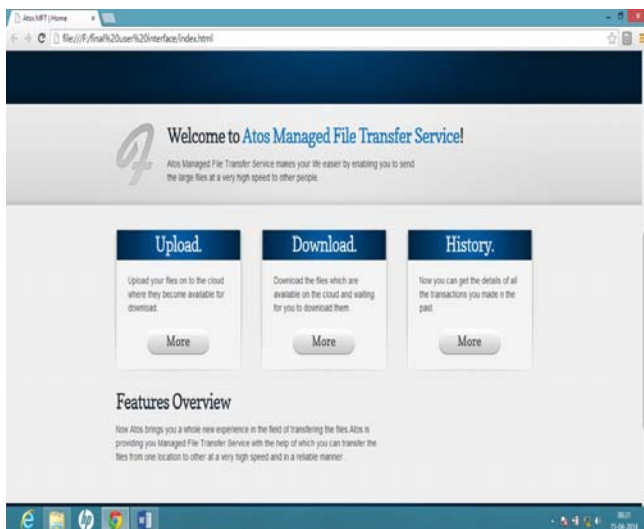


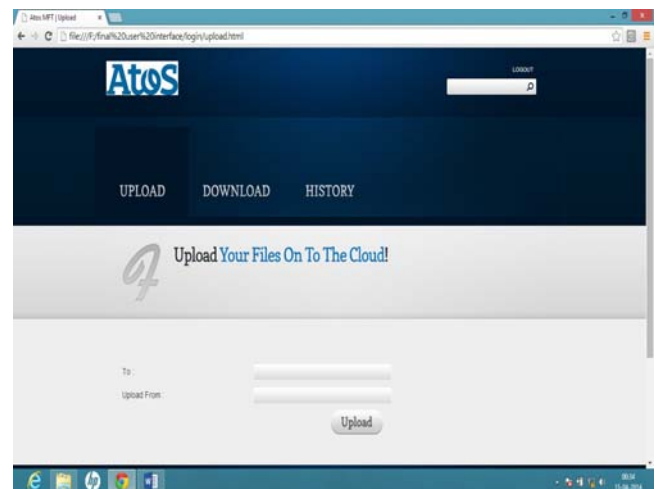
Figure 1: Working of Cloud based MFT

Due to certain network problems if the network is broken during the transfer of data, the user do not need to transfer the data from scratch. The data can be downloaded from where it was broken with the resume functionality. Multiple clients can download and upload the files at the same time. The user will access the user account to see logs of previous data transfer and also billing options.

The implementation snapshots are as shown below



An account must be created by each user in order to use the MFT services.



Above Snapshot is of upload page. To field will have the id of the person for whom the file is to be uploaded on the cloud. Only specified person will be able to view the files on the cloud. Upload from field will contain the path from where the file is to be selected.



The download page is as shown above. The file id will be sent to the user via email once the file is uploaded for that person. Even the person will be capable of viewing the files available on the cloud. Download path will contain the path where the data is to be downloaded on the client's machine.

#### IV. UDT

UDT is a connection-oriented duplex protocol. It supports both reliable data streaming and partial reliable messaging. Fig. 2 describes the relationship between the UDT sender and the receiver. In Fig. 2, the UDT entity A sends application data to the UDT entity B. The data is sent from A's sender to B's receiver, whereas the control flow is exchanged between the two receivers.

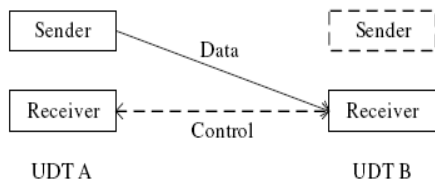


Figure 2: The relation between UDT sender and UDT receiver

The receiver is also responsible for triggering and processing all control events, including congestion control and reliability control, and their related mechanisms as well.

UDT uses rate-based congestion control (rate control) and window-based flow control to regulate the outgoing data traffic. Rate control updates the packet-sending period every constant interval, whereas flow control updates the flow window size each time an acknowledgment packet is received.

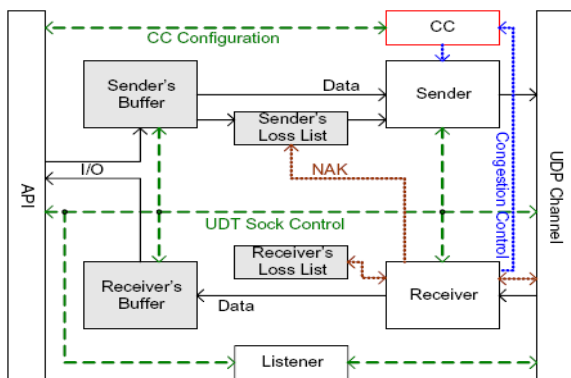


Figure 3: Software Architecture Of UDT implementation

Fig. 3 depicts the UDT software architecture. The UDT layer has five function components: the API module, the sender, the receiver, the listener, and the UDP channel, as well as four data components: sender's protocol buffer, receiver's protocol buffer, sender's loss list, and receiver's loss list.

Because UDT is bi-directional, all UDT entities have the same structure. The sender and receiver in Fig. 3 have the same relationship as that in Fig. 2. The API module is responsible for interacting with applications. The data to be sent is passed to the sender's buffer and sent out by the sender into the UDP channel. At the other side of the connection (not shown in this figure but it has the same architecture), the receiver reads data from the UDP channel into the receiver's buffer, reorders the data, and checks packet losses.

Applications can read the received data from the receiver's buffer. The receiver also processes received control information. It will update the sender's loss list (when a NAK is received) and the receiver's loss list (when loss is detected). Certain control events will trigger the receiver to update the congestion control module, which is in charge of the sender's packet sending.

The UDT socket options are passed to the sender/ receiver (synchronization mode), the buffer management modules (buffer size), the UDP channel (UDP socket option), the listener (backlog), and CC (the congestion control algorithm, which is only used in Composable UDT). Options can also be read from these modules and provided to applications by the API module.

#### V. CONCLUSION

In this report we have represented The cloud based managed file transfer services as a web based application and we are going to provide it as a SaaS to the clients. The Managed File Transfer Services is the need of today as MFT Services can quickly and cost-effectively deliver large files to anyone, anywhere – regardless of platform. This product will be able to handle large files i.e. upload to cloud, download from cloud to client and be able to resume download in case of network failure. The User Interface will be very User Friendly and interactive. It will provide a secure, Reliable, Fast way of transferring large files and keeping track of them. The product can keep the information updated all the time so that the clients get the updated information all the time.

#### REFERENCES

- [1] Yunhong Gu, Robert L. Grossman, "UDT: UDP-based data transfer for high-speed wide area networks" National Center for Data Mining, University of Illinois at Chicago, 851 S Morgan St, M/C 249, Chicago, IL 60607, United States
- [2] Yunhong Gu and Robert L. Grossman, "Supporting Configurable Congestion Control in Data Transport Services", Proceedings of the 2005 ACM/IEEE SC'05 Conference (SC'05).