N.B. Please activate comments in word to see more information

Paper Title
Paper Subtitle

Author1 LastName1
Affiliation
mail@youraftiliation.com

Author2 LastName2
Affiliation
mail@youraftiliation.com

Author 3 LastName3
University of ABC
Author3.LastName3@abc.edu

Author4 LastName4
University of ABC
Author4.LastName4@abc.edu

Author5 LastName5
Affiliation
mail@youraftiliation.com

Abstract
This is the abstract. It should not be longer than 200 words. Please make sure it is in italics, 12 pt Times New Roman. This is the abstract. It should not be longer than 200 words. Please make sure it is in italics, 12 pt Times New Roman. This is the abstract. It should not be longer than 200 words. Please make sure it is in italics, 12 pt Times New Roman. This is the abstract. It should not be longer than 200 words. Please make sure it is in italics, 12 pt Times New Roman. This is the abstract. It should not be longer than 200 words. Please make sure it is in italics, 12 pt Times New Roman. This is the abstract. It should not be longer than 200 words. Please make sure it is in italic's, 12 pt Times New Roman. This is the abstract. It should not be longer than 200 words. Please make sure it is in italics, 12 pt Times New Roman. This is the abstract. It should not be longer than 200 words. Please make sure it is in italics, 12 pt Times New Roman.

Keywords: Keyword1, Keyword2, Keyword3, Keyword4, keyword5.

Paper Title
Paper Subtitle

1. Introduction

Data transfers constitute one of the most commonly performed tasks in the Internet today. Users publishing files to a website, moving files that are too large for an email application, transferring files securely between different computers, uploading photos to services such as Flickr (www.flickr.com) or backing up a to a remote server are all engaged in data transfer operations. File Transfer Protocol (FTP), for example, is still used to perform bulk data transfers across networks (Grzywa et al., 2001). It is important to note that this simple-sounding task (data transfer) is essential for the interoperation of networked computers, especially when the different computing environments of modern heterogeneous networks are taken into account.

Potential user devices in the network range from Personal Digital Assistants (PDAs) and mobile phones to specialized servers and large-scale corporate systems. All of these devices could operate using different word formats; they might store the data in different forms and forward packets in non-compatible ways (big-endian vs. little-endian computers) (Tanenbaum, 2003).

2. Overview of Data Transfer Mechanisms

There are two main approaches to perform data transfers in today's IP-based networks. One is based on FTP and several variations to that protocol and the second, and much newer, approach is based on peer-to-peer networking configurations. Let's review both of them.

2.1 FTP Overview

Essentially, FTP is a client-server protocol that facilitates the transfer of files between two computers connected via a network, such as the Internet (Brown & Jaatun, 1992). The protocol has several fundamental components that perform the required functions for establishing the sessions between systems, and coordinates the transfer of the files. These components are as follows:

- Server-FTP that consists of the Server Protocol Interpreter and the Server Data Transfer Process;
- Server-side File System;
- User-FTP that consists of:
    - the User Interface,
    - User Protocol Interpreter, and
    - the User Data Transfer Process;
- User-side File System;
- A User; and
- A communication channel for the control and data connections.

An alternative configuration of FTP provides for the transfer between two servers through a third system that provides the control function between the servers. In both of the configurations, FTP requires that the control channel be open during the data transfer process (Postel & Reynolds, 1985).

The FTP architecture is displayed below (see Figure 1):

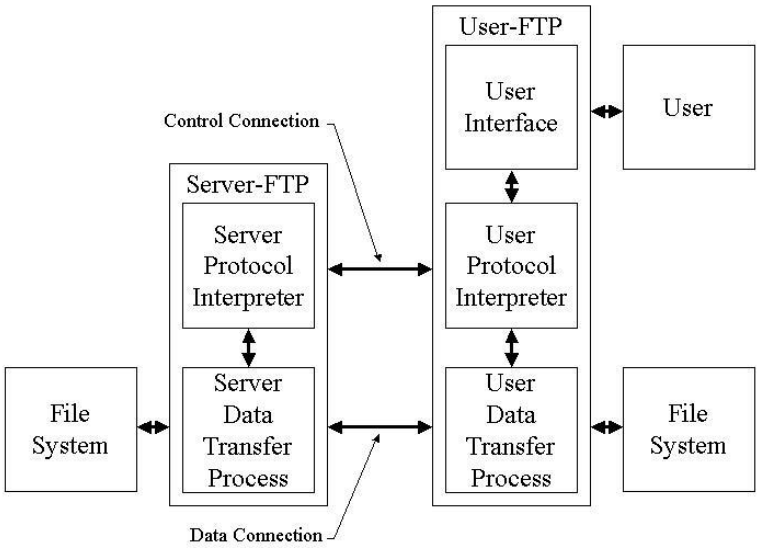

Figure 1: FTP Architecture
(Source: Postel & Reynolds, 1985)

2.1.1 Telnet Authentication and Encryption

As previously discussed, FTP utilizes the control connection to coordinate the data connection and execute commands on the File Systems of both the user and the server. FTP utilizes the Telnet protocol to execute the commands (Postel & Reynolds, 1985). This fundamental design lends itself to security breaches that may permit eavesdropping of user ID's, passwords, file names, and other information passed through the control channel. It also may allow an active attacker to change settings and execute commands on the file system (Brown & Jaatun, 1992). This fundamental security flaw was initially addressed when Borman (1993) proposed the passing of authentication information, and a mechanism to enable encryption of the data after successful authentication for the Telnet protocol. The result was that user passwords would not be sent in clear text, and the data stream would be encrypted utilizing any general authentication and encryption system. However, it should noted that the "Telnet authentication and encryption option does not provide for integrity protection only (without confidentiality), and does not address the protection of the data channel" (Horowitz & Lunt, 1997).

Scenarios				
Send Request Response	Request Hash	Storage Hash	Generated Hash	Action Required / Notification Returned
	h_1	h_1	h_1	Validated file, integrity confirmed, send file
	h_1	h_1	h_2	File has changed since original message, original hash may not have been updated, confirm file integrity before continuing, regenerate hash and reconfirm
	h_1	h_2	h_2	Message to Recipient is invalid or out of date, may be invalid request, request retransmission of message
	h_1	h_2	h_1	Original hash may be corrupted, update hash file, confirm before transmitting file
	h_2	h_1	h_1	Message to Recipient is invalid or out of date, issue new notification, request retransmission of message
	h_2	h_2	h_1	File has changed since original message, original hash may not have been updated, confirm file integrity before continuing, regenerate hash and reconfirm
	h_2	h_1	h_2	Original hash may be corrupted, update hash, confirm before transmitting file
	h_2	h_1	h_3	Re-evaluate all controls for file, discontinue distribution of file, and everyone is going to hell in a hand-basket ☺

Table 1: Send Request Response

5. Conclusion

A basic requirement by users is that files be transferred efficiently and without modification. File integrity, therefore, becomes a critical element in the sharing of files. During the transfer of a file, a self-correcting system would permit the user to acquire a file while the error controls would operate transparently in the background. An added feature of the Hash Triplet is that it provides the system with specific points of origins for the root error causes and allows system designers to promote a wider range of remedies.

5.1 Formatting References

Book. The format is: Single Space, Hanging 5mm: Author(s) (date). *Book Title in Italics*, *edition in italics* (if appropriate). Place of Publication: Publisher, pages (if appropriate)

Journal Article. The format is: Single Space, Hanging 5mm: Author(s) (date). "Title." *Journal Name in Italics*, *Volume in italics*(Issue), pages

Edited Book. The format is: Single Space, Hanging 5mm: Editors(s) (ed.) (date). *Book Title in Italics*, Publication: Publisher, pages (if appropriate)

Article in Edited Book. The format is: Single Space, Hanging 5mm: Authors(s) (date). "Title of Article." In Editor name (ed.) Book *Title in Italics*, Place of Publication: Publisher, pages

Website Reference. The format is: Single Space, Hanging 5mm: Authors(s) (date). *Article Title in Italics*. Retrieved 30 Month 2019, from www.website.com/article

The references should look like the samples below.

References

Burgess, M. (2019, January 21). *What is GDPR? The Summary Guide to GDPR Compliance in the UK*. Retrieved 26 January 2018, from https://www.wired.co.uk/article/what-is-gdpr-uk-eu-legislation-compliance-summary-fines-2018

Coleman, D. and R. Kanna (eds.) (1995). *Groupware Technology and Applications*, Upper Saddle River, NJ: Prentice Hall PTR.

McNurlin, B. and R. Sprague (1998). *Information Systems Management in Practice*, *6th edition*, Upper Saddle River, NJ: Prentice Hall, 133-170.

Nunamaker, J. F., R. O. Briggs, and D. D. Mittleman (1995). "Electronic Meeting Systems: Ten Years of Lessons Learned." In Coleman, D. and R. Kanna (eds.) *Groupware Technology and Application*s, Upper Saddle River, NJ: Prentice Hall PTR, 146-193.

Singh, S. and I. Chana (2016). "A Survey on Resource Scheduling in Cloud Computing: Issues and Challenges." *Journal of Grid Computing*, *14*(2), 217-264.

Sundt, C. (2006). Information Security and the Law. *Information Security Technical Report,* 11 (1), 2-9.

Svantesson, D. and R. Clarke (2010). "Privacy and Consumer Risks in Cloud Computing." *Computer Law and Security Review, 26*(4), 391-397.

Wei, L., H. Zhu, Z. Cao, X. Dong, W. Jia, Y. Chen, and A. V. Vasilakos (2014). "Security and Privacy for Storage and Computation in Cloud Computing." *Information Sciences, 258*, 371-386.

Appendix 1

DON'T FORGET!
• Remove reviewing changes (track changes) before submission.
• Follow the format and numbering of paper content exactly as described and shown in the paper guide.
• Include all the tables as MS-Word tables, not as in jpeg or similar format.