The Investigation of Communications Protocol

Zihuai Ma

Tianjin Foreign Languages School Affiliated to Tianjin Foreign Studies University, Tianjin, 300230, China
championfengzi@tju.edu.cn

Abstract. Despite the long-standing presence of the internet in our lives, there remains a significant portion of the population unaware of how their data is transmitted, the speed of their connections, and the security of their online activities. It is essential to consolidate knowledge in this field, enabling not only everyday users to understand the mechanics of their online interactions for troubleshooting purposes but also to equip professionals with accurate insights into this realm. In pursuit of this goal, the author conducted experiments and conducted extensive online research. Through a comparative analysis of various protocols, two emerge as particularly valuable and widely employed: Secure Shell (SSH) and Hypertext Transfer Protocol (HTTP). SSH, renowned for providing remote shell access in Linux environments, exhibits exceptional scalability, making it suitable for file transmission and even for launching desktop software. For instance, applications like XShell facilitate running graphical software on Linux servers, akin to remote desktop functionality. HTTP, on the other hand, is a versatile protocol capable of web browsing, file downloads, media streaming, and serving as an effective management tool. Much like SSH, HTTP enables web server management through web-based dashboards, often referred to as "web backstages." Moreover, it facilitates file uploads and provides tools for establishing shell connections, allowing users to efficiently manage their websites through the command line interface. These protocols, SSH and HTTP, represent essential components of modern internet communication and management.

Keywords: Communication Protocol, HTTP, SSH

1 Introduction

The field of internet communication relies fundamentally on Communication Protocols, representing a cornerstone of contemporary society [1-3]. These protocols serve as standardized systems facilitating the transmission of information among two or more terminals via various physical mediums, establishing a shared language for computer and online device communication. Communication protocols delineate the structure, semantics, synchronization mechanisms, and error detection and correction procedures essential for effective communication. Such protocols can be implemented in hardware, software, or a combination of both. Efficient information exchange within communication systems necessitates the adoption of a common format or protocol. Each transmitted message carries precise significance, enabling the intended recipient to respond accordingly and autonomously, adhering to predetermined response behaviors. It is

B. H. Ahmad (ed.), *Proceedings of the 2023 International Conference on Data Science, Advanced Algorithm and Intelligent Computing (DAI 2023)*, Advances in Intelligent Systems Research 180,
https://doi.org/10.2991/978-94-6463-370-2_58

imperative that all participating entities mutually agree upon the communication protocol to ensure its successful implementation and operation. In essence, communication protocols form the linchpin of modern internet-based communication systems, enabling the seamless flow of data and information. Attaining a consensus necessitates the presence of technical standards within an agreement. Similarly, programming languages should possess corresponding standards concerning computation. Drawing an analogy, programming languages fulfill the role of structured computation, while communication protocols serve the purpose of facilitating seamless interactions. Multiclass protocols encompass various facets of a unified data transmission process, encompassing the concurrent utilization of distinct protocol modules and protocol stacks implemented in software. The development of Internet communication protocols falls under the purview of the Internet Engineering Task Force (IETF). Wired and wireless transmission standards are overseen by the Institute of Electrical and Electronics Engineers (IEEE), while the International Organization for Standardization (ISO) assumes responsibility for other protocol categories. Furthermore, the International Telecommunication Union-Telecommunication Standardization Sector (ITU-T) is tasked with regulating telecommunications transmission and defining the format for the Public Switched Telephone Network (PSTN). These governing bodies collectively establish the framework and standards that underpin various facets of modern communication protocols.

Today, with the integration of public switched telephone networks and network technology, the situation is driving further convergence and consolidation of communication standards. Common network protocols include Transmission Control Protocol/ Internet Protocol (TCP/IP), Secure Shell (SSH), File Transfer Protocol (FTP), Hypertext Transfer Protocol (HTTP), Telnet, Post Office Protocol - Version 3 (POP3), Secure Socket Layer/ Transport Layer Security (SSL/TLS) etc [4-10]. TCP/IP stands as the preeminent communication protocol in contemporary networking. Also referred to as Transmission Control Protocol/Internet Protocol or simply network communication protocol, TCP/IP constitutes the foundational communication protocol within network usage. It meticulously outlines the standards and methodologies governing communication across various facets of the Internet. Furthermore, TCP/IP encompasses two pivotal protocols that ensure the expeditious and comprehensive transmission of network data information. Given its ubiquitous presence and essential role in everyday life, an exhaustive review of this domain is warranted. Such a review would encompass a comprehensive introduction to the protocols, a thorough examination of their advantages and disadvantages, and a comprehensive analysis. Ultimately, this article serves to provide a holistic overview of the subject matter at hand.

2 Protocol Method

2.1 TCP/IP

The TCP/IP protocol framework comprises four distinct layers, each serving specialized functions:

Application Layer: Positioned as the initial layer of the TCP/IP protocol stack, the application layer assumes the crucial role of directly furnishing services to application processes.

Transport Layer: Serving as the second stratum within the TCP/IP protocol, the transport layer assumes a pivotal position throughout the entire TCP/IP architecture. Within this layer, both Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) play integral roles.

Network Layer: The network layer is situated at the third tier within the TCP/IP protocol hierarchy. In the TCP/IP protocol context, the network layer undertakes functions such as the establishment and termination of network connections and the facilitation of IP address lookup.

Network Interface Layer: In the TCP/IP protocol, the network interface layer occupies the fourth and final layer. Due to its amalgamation of the physical layer and data link layer, the network interface layer not only serves as the conduit for data transmission but also provides precise routing capabilities for the network layer.

2.2 SSH

SSH stands for Secure Shell. Secure Shell Protocol is a security protocol built on the application layer. By encrypting and verifying passwords, it can provide a secure transmission environment for network services in an insecure network, enabling the connection between SSH clients and SSH servers. Therefore, SSH is based on the client server mode.

2.3 FTP

The File Transfer Protocol (FTP) consists of two integral constituents: the FTP server, tasked with the storage of files, and the FTP client, which facilitates user access to resources hosted on the FTP server through the FTP protocol. In the context of website development, the FTP protocol is frequently employed to effectuate the transfer of web pages or applications to a web server. Furthermore, owing to its notable efficiency in data transmission, FTP is typically the protocol of choice for transferring substantial files across computer networks.

2.4 HTTP

HTTP is an application layer protocol used to exchange data between web browsers and web servers. Through HTTP, web browsers can send requests and obtain responses from web servers, thereby enabling access and transmission of web pages. HTTP uses TCP as the transport layer protocol and adopts a request response model for communication. Within HTTP, a request message encompasses a request line, request header, and request body. Conversely, a response message comprises a status line, response header, and response body. The HTTP protocol's design goals prioritize simplicity, scalability, and strong interoperability.

2.5 Telnet

The Telnet protocol, belonging to the TCP/IP protocol suite, serves as the standard and principal method for Internet-based remote login services. It empowers users with the capability to execute tasks on a remote host from their local computing devices. This

functionality is achieved by employing the telnet program on the end user's computer to establish a connection with the server. Subsequently, end users can input commands into the telnet program, which will execute on the server, akin to direct input on the server's console. This method affords local control over the server. To initiate a telnet session, the provision of a username and password is obligatory to gain access to the server. Telnet is widely employed for the remote administration and management of web servers within the realm of computer networking.

2.6 POP3

Post Office Protocol version 3 is a standard protocol for email, which defines how email clients obtain emails from email servers. The POP3 protocol is a straightforward, stateless protocol that allows users to download emails from email servers through email clients such as Outlook, Thunderbird, etc. for offline viewing and processing.

2.7 SSL/TLS

The SSL protocol stands as an internationally recognized standard for encrypting communication and verifying identities. Alongside its successor, Transport Layer Security (TLS), SSL serves as a security protocol that ensures the confidentiality and data integrity of network communications. TLS and SSL operate by encrypting network connections spanning from the transport layer to the application layer. Positioned between the TCP/IP protocol and various application layer protocols, SSL furnishes a secure framework for data transmission.

The SSL protocol comprises two distinct layers:

SSL Record Protocol: Positioned atop reliable transmission protocols like TCP, this layer facilitates fundamental functions such as data encapsulation, compression, and encryption for higher-level protocols.

SSL Handshake Protocol: Situated atop the SSL Record Protocol, the SSL Handshake Protocol assumes the critical role of identity authentication, negotiation of encryption algorithms, and the exchange of encryption keys between communication entities before the commencement of actual data transmission.

3 Analysis of advantages of disadvantages

This part will show some protocols which are used most frequently.

3.1 TCP/IP

The advantages of TCP/IP can be analyzed in many aspects. Some important parts are provided.

1.Standardization and openness: TCP/IP protocol is a widely used and open network protocol, and its standardization and openness have made it widely used and promoted globally.

2. Layered structure: TCP/IP protocol is a layered protocol composed of four layers, each with different functions and tasks. It can be flexibly combined and disassembled, making it easy to adapt to diverse network environments and application scenarios.

3. Reliability and Performance: The transport layer TCP protocol of TCP/IP protocol provides highly reliable data transmission services, ensuring data integrity and sequence, and has mechanisms such as flow control and congestion control. At the same time, it performs faster than the UDP protocol and is suitable for situations where the speed is faster, but the data does not need to be guaranteed.

4. Routing function: The network layer of TCP/IP protocol adopts the IP protocol, which has routing function. It can identify different networks and hosts, and select routes and paths for data packets. Routing algorithms can make the network scalable and adaptable according to different loads and network topology structures.

5. Flexibility and Scalability: The TCP/IP protocol has strong flexibility and scalability, which can easily adapt to different development needs and network application environments. The TCP/IP protocol also supports various application layer protocols, such as HTTP, DNS, SMTP, FTP, etc., which can meet diverse network service requirements.

3.2 SSH

The conventional three-layer architectural paradigm embodies the fundamental Model, View, and Control (MVC) concept, offering developers a structured approach to addressing complex problem-solving tasks. This architectural framework aids in mitigating the challenges associated with developing solutions for intricate problems, fostering agility in the adaptation to new requirements, and ultimately leading to reduced development timelines and costs.

Robust Scalability: Prominent SSH technology enjoys robust support from a vibrant user community, endowing it with notable scalability and exceptional plug-and-play capabilities tailored for specialized applications. This inherent flexibility effectively mitigates the technical constraints that might impede the attainment of specific functionalities, ensuring a high degree of adaptability.

Effective Maintainability: Business systems frequently evolve with emerging requirements. The three-layer architecture offers an effective approach to minimize the potential risks associated with requirement modifications by logically segregating the presentation and logic layers. As technologies advance or systems age, there may arise a need for system refactoring. In such instances, the SSH architectural framework exhibits a significantly higher success rate for refactoring efforts compared to alternative architectural models, rendering it a pragmatic choice for long-term system maintainability and evolution.

3.3 FTP

1. Convenient and fast file transfer

The FTP server allows users to easily transfer files. The files in the root directory of the FTP server's website can be replaced, and then transfer the files to the Internet through the FTP client.

2. Wide application range

FTP servers have a wide range of applications, not only for ordinary users to upload and download files, but also for enterprises to use in file sharing, backup, collaboration, and other work.

3. High safety

The FTP server uses Secure Transfer Protocol (SFTP) or Secure Sockets Layer (SSL/TLS) for encrypted transmission of data. By adopting this method, the security and legality of data transmission can be ensured, thereby avoiding data leakage and theft.

3.4 HTTP

1. It is a reliable data transmission protocol (the underlying TCP protocol ensures the correctness of the transmission content order and ensures that packets are not lost during transmission).
2. Simple and fast. When a client requests services from the server, it only needs to transmit the request method and path.
3. Flexibility. HTTP allows the transmission of any type of data object.
4. It has statelessness.

4 Conclusion

This paper comprehensively explores approximately ten key internet protocols, each possessing unique strengths and contributing significantly to internet communication. Let's delve into a couple of the most exemplary protocols. Linux Operating System users, particularly internet engineers, frequently opt for SSH due to its exceptional scalability and maintainability. SSH boasts versatile utility, offering a multitude of functions and ease of updates. Its three-layer architecture minimizes the risks associated with modifications. Similarly, HTTP enjoys widespread use, serving as the primary protocol for fetching content from web servers. It boasts simplicity, requiring nothing more than a web browser for access, enabling users to browse diverse data objects effortlessly. Nevertheless, as the internet evolves, it can be anticipated the emergence of new communication protocols that may surpass the speed and security of today's standards. However, it is likely that these new protocols will be rooted in TCP/IP or similar frameworks. Hence, acquiring a solid understanding of TCP/IP protocols remains a fundamental necessity for navigating the ever-evolving landscape of internet communication.

References

1. Al-Sarawi, S., Anbar, M., Alieyan, K., et al.: Internet of Things (IoT) communication protocols, 2017 8th International conference on information technology (ICIT). IEEE, 685-690, (2017)
2. Gouda, M. G., Multari, N. J.: Stabilizing communication protocols. IEEE Transactions on Computers, 40(04): 448-458, (1991).
3. Merlin, P.: A methodology for the design and implementation of communication protocols. IEEE Transactions on communications, 24(6): 614-621, (1976).
4. Forouzan, B. A.: TCP/IP protocol suite. McGraw-Hill Higher Education (2002).
5. Parziale, L., Liu, W., Matthews, C., Rosselot, N., Davis, C., Forrester, J., Britt, D. T.: TCP/IP tutorial and technical overview, (2006).

6. Barrett, D. J., Silverman, R. E.: SSH, the Secure Shell: the definitive guide. O'Reilly Media, Inc (2001).
7. Ylonen, T., Lonvick, C.: The secure shell (SSH) protocol architecture (No. rfc4251) (2006).
8. Housley, R., Hoffman, P.: Internet X. 509 public key infrastructure operational protocols: FTP and HTTP (No. rfc2585), (1999).
9. Bellovin, S.: Firewall-friendly FTP (No. rfc1579), (1994).
10. Seufert, M., Egger, S., Slanina, M., Zinner, T., Hoßfeld, T., and Tran-Gia, P.: A survey on quality of experience of HTTP adaptive streaming. IEEE Communications Surveys & Tutorials, 17(1), 469-492, (2014).