

Python TCP/IP libraries: A Review

Ghassan Muslim Hassan¹, Nadia Mahmood Hussien¹, Yasmin Makki Mohialden¹

Email: gmhalsaddi@uomustansiriyah.edu.iq

¹Computer Science Department, College of Science, Mustansiriyah University, Baghdad-Iraq

Abstract

The Internet's core is TCP/IP, which stands for Transmission Control Protocol/Internet Protocol. It connects network devices on the internet via communication protocols. Python has several TCP/IP packages due to its popularity and flexibility. This paper describes the most popular Python libraries for TCP/IP protocol implementation, including socket, asyncio, Twisted, and Scapy. To help developers choose a library, we compare its benefits, cons, and areas of use, including criteria other than speed and memory utilization. When making web apps, choose wisely.

Keywords: TCP/IP, Asyncio, Twisted, Scapy, Protocol implementation, Python libraries

Received : April 7, 2023

Received in Revised: May 5, 2023

Accepted: May 18, 2023

Introduction

TCP and higher-level protocols are executed on the host system. Major computers will use network software and TCP and Internet protocols to add some computer systems to the network. If both the host computer and the main computer employ the correct protocol for communication, the TCP standard may work for the main computer (FOROUZAN, 2002).

For working with TCP/IP protocols, there are many Python tools to choose from. Here are four famous choices and some things to think about when choosing between them. A socket library is a low-level tool that lets sockets be made and managed. It works with both TCP and UDP protocols and can be used for both client and server applications. But programming a socket can be hard and error-prone, especially when it comes to contacts that happen at different times. Reference (<https://docs.python.org/3/library/socket.html>).

Twisted is an event-driven networking framework that lets you build network apps with a high-level API. It works with many different protocols, like TCP, UDP, HTTP, and SMTP. It also has features like asynchronous I/O and SSL/TLS protection. The learning curve is steep, though, so it might not be the best choice for simple (<https://twisted.org/>).

Asyncio is a Python library that is built-in and provides a structure for writing programs that run in parallel. It works with both TCP and UDP protocols and can be used for both client and server applications. It's perfect for apps that need to handle a lot of contacts at once, but it might be too much for simpler projects (<https://docs.python.org/3/library/asyncio.html>).

Scapy is a Python tool that lets you change and analyze packets. It makes it possible to make custom TCP/IP packets, simulate network attacks, and look at network data. But it might not be the best way to make full-fledged network applications (<https://scapy.net/>).

The goal of this paper is to show the main features of each Python library that supports TCP/IP protocols. Think about which protocols you require and how effectively they work when choosing one of these libraries.

Structure of the paper: Section 2: Related Work, Section 3: Libraries Overview and Comparison, and Section 4: Conclusion.

Related work

Several systems have been developed based on the TCP/IP protocol using Python and its library. Some examples are:

In 2018, Sandia assessed a remote grid-monitoring gateway and communication-enabled PV inverter for cyber security. The goal was to inform the solar sector, grid operators, and government about present hazards and provide community-based remedies. The team detected some security flaws and some well-designed features. These findings led to security recommendations for DER devices (Carter et al., 2018)

In 2020, Using Python 3.7, a team studied distributed networks, specifically blockchain networks, and created a peer-to-peer electricity trading system. This blockchain network needs RSA encryption, several master nodes, multiple client nodes, and conflict resolution (ARNAO et al., 2020).

In 2020, BEDI, Punam, and DUAs Arti presented a local area network covert channel employing the Overflow field of the Timestamp option of Internet Protocol, version 4. This storage-based network steganography uses timestamps for network debugging and measurement. They used genuine overflow field values, making clandestine communication difficult to detect (BEDI 2020).

In 2021, Using a TCP-based Python library and a server, a team published a paper on how to develop a chat application. This chat application is useful even if several clients are connected because parallel programming uses threads to transmit and receive messages in real time (SALIHU, 2021).

HTNEN, Seppo, et al. discussed the main obstacles to seamless multi-connectivity in the same year. MULTI, a session-layer solution, can handle some of this problem's important subproblems. MULTI's TCP overhead was less than 5% compared to the native asyncio Python module (HÄTÖNEN, 2021).

In 2022, A team created a Python client for the Isabelle server, allowing researchers and students to interface with the server over TCP using Python scripts. This avoids the difficulties of combining the Python script with Isabelle's development languages (ML and Scala). They also described additional features since the client's first release a year ago with Isabelle's ML and Scala development languages. They also described additional features that have appeared since the client's first version was announced a year ago (SHMINKE, 2021).

Libraries Overview and Comparison

The Python library or module that supports TCP/IP protocols are: -

Socket

Python provides good coverage of low-level networking interfaces. It all starts with the BSD socket interface. Python has a socket module that provides the functionality needed to use the socket interface. Network programming in any programming language can start with sockets

Sockets are the most popular form of inter-process communication, especially for cross-platform communication. They were invented by Berkeley and are part of BSD Unix. By combining sockets with INET, communication between machines is made very easy.

Sockets are essentially the endpoints of a two-way communication link. Identification by IP address and port number. Sockets should be properly configured on both ends so that they can initiate connections, listen for incoming messages, and then send responses to both ends of the client-server environment (Sarker, 2015).

Twisted

Twisted is an event-based web application framework that supports Python 3.6+. It contains modules for many different purposes. Twisted is a library that allows you to easily create web applications in Python. While Twisted manages to abstract most of the low-level networking details, at least having a broad understanding of network I/O is very helpful. When you connect two processes at a very simple level using mechanisms that allow them to exchange data, you network them. It helps to imagine two physical machines connected by a single wire. Once the two machines are connected with a single cable, they can use it. A machine can write data at one end of its line, and some time later it reaches the other end of the line. Additionally, data can arrive at the same end of the line at all times(Fettig, 2005). .

Asyncio

The asyncio standard library module is new in Python 3.4 and attempts to bring some standardization of asynchronous I/O into the standard library. The Asyncio library uses a coroutine-based programming style. It provides a powerful loop class to which our program can send prepared tasks (called coroutines) to run asynchronously. The event loop is responsible for scheduling tasks and optimizing performance around blocking I/O calls. It has built-in support for socket-based networking, making building a basic server an easy task (Sarker, 2015).

Scapy

Scapy is a library supported by both Python2 and Python3. It is used to interact with packets on the network. It has several functions that allow us to easily forge and manipulate bags. Using Scapy modules, we can create various network tools such as ARP spoofers, network scanners, packet dumpers, etc. This module can be used to create more advanced cybersecurity and ethical hacking tools.

A network scan is all about scanning the entire network we are connected to and finding out which clients are connected to our network. We can identify each customer based on their IP and MAC address. Through ARP ping, we can find out the active systems in the network (Rohith et al., 2018).

Comparative analysis of libraries The primary comparison displayed in Table 1 is (Lubanovic, 2022; Soni, 2021; Fettig, 2005; Thaker, 2020, Hattingh, 2020; Rohith et al., 2018).

Table 1. Comparative analysis of libraries

Factors	Socket	Twisted	Asyncio	Scapy
General	Python's Socket package provides low-level TCP/IP. Its API simplifies socket generation, server connections, and data transfer.	A high-level Python event-driven networking engine for scalable, asynchronous network apps.	Python's built-in asynchronous library allows concurrency. Networking simplifies non-blocking	Packet manipulation. Creates packets, sniffs communications, and evaluates protocols. Though not a

	Network programming is popular because Socket is cross-platform.	Supporting protocols and providing event abstractions simplifies networking code. Simpler sockets.	apps. The socket lacks coroutines and a contemporary API.	networking library, network engineers and developers need it.
Module	Python's built-in libraries, socket and asyncio, are always available. Twisted and Scapy are third-party libraries. PyPI's pip installs them.			
Availability	<ul style="list-style-type: none"> • Socket is the simplest networking library. • Twisted is more powerful but more difficult for asynchronous network applications. • Asyncio simplifies concurrent network application development. • Scapy analyzes packets. 			
Project Complexity	Simple client-server applications may use socket. Twisted or asyncio may simplify complex applications.			
API for Using TCP/IP Protocols	<ul style="list-style-type: none"> • Socket simplifies TCP/IP. Networking is simple. As networking needs rise, socket requires developers to control low-level properties like buffering and flow management, making it harder to utilize. • Twisted's extensive API allows for complicated networking activities. Developers may take longer to master it than Socket. • Asyncio facilitates concurrent network application development. Routines facilitate network request processing. Mastering async and asynchronous code structures takes time. • Scapy's robust interface helps network engineers and developers build and control network packets. As a non-networking library, network connectivity and protocol implementation may be less beneficial. 			
Easy of Programming	<ul style="list-style-type: none"> • The simplest library, Socket, offers few functionalities. • Twisted and Asyncio provide more complicated abstractions and capabilities but a steeper learning curve. • For advanced network research, the most difficult library, Scapy, manipulates packets. 			
Download	All four libraries are free. Twisted and Scapy must be downloaded; however, Python's standard library includes socket and asyncio. Each library's online website has documentation, tutorials, and community support for developers.			
Web Sites	https://docs.python.org/3/library/socket.html	https://twisted.org/	https://docs.python.org/3/library/asyncio.html	https://scapy.net/
Application Type	Scapy, Twisted, Asyncio, and Socket are used for network testing, monitoring, and forensics. It depends on the use case and features.			
Popularity	Asyncio and Scapy are becoming more popular due to their performance and usability.			

Use of Memory	<ul style="list-style-type: none"> ● Socket programming is easier with low-level libraries. For simple networking tasks, it's memory-efficient. ● Networking library asynchronous High-level socket programming with Asyncio Coroutines and event loops consume more memory than asynchronous programming. ● Asyncio faces Twisted. Networking engine. Libraries with higher-level functionality require additional RAM. ● Custom network packets are sent by Scapy. RAM handles packet parsing and complex functions.
Speed	<ul style="list-style-type: none"> ● Low-level library Sockets simplify socket programming. It is the fastest networking library for modest data transfers. ● Asynchronous networking library Asyncio performs better when handling many connections. Coroutines and event loops can optimize resource use and performance. ● Twisted is another high-traffic asynchronous networking library. Its high-level features and built-in protocols can handle sophisticated networking tasks. ● Scapy manipulates packets. Complex features and parsing skills slow data processing.

Conclusion

TCP/IP protocols govern computer communication, routing, and network connections. Research institutions, universities, governments, and businesses use it to communicate online. TCP/IP makes a machine an Internet host that can connect to others. Python is an object-oriented, dynamically semantic programming language with built-in high-level data structures, dynamic typing, and dynamic binding. Python has many system development libraries. This page recommends TCP/IP libraries depending on developer needs and efficiency. Internet communication is based on TCP/IP. Python has several TCP/IP libraries due to its flexibility and popularity. This paper discusses socket, asyncio, Twisted, and Scapy, the most popular Python TCP/IP libraries. We help developers choose a library based on benefits, constraints, and applications (memory use, availability, speed, and more). Web app development requires smart choices. All the libraries are used for network testing, monitoring, and forensics. It depends on the use case and features. Asyncio and Scapy are more popular due to their performance and usability.

Acknowledgments

Describe The Authors would like to thank Mustansiriyah University, Baghdad–Iraq (<https://uomustansiriyah.edu.iq/>) for its support in the present work.

References

- Arnao, Vicente; Luis Fujdiak, Radek. (2020). Desarrollo de un sistema Peer-to-Peer (P2P) basado en Blockchain para redes de comercio de energía.
- Bedi, Punam; Dua, Arti. (2020). Network steganography using the overflow field of timestamp option in an IPv4 packet. *Procedia Computer Science*, 171: 1810-1818.
- Carter, C., Cordeiro, P. G., Onunkwo, I., & Johnson, J. T. (2018, January). Cyber assessment of distributed energy resources. *In Conference Record of the IEEE Photovoltaic Specialists Conference* (Vol. 2017, No. SAND2018-0281C). Sandia National Lab.(SNL-NM), Albuquerque, NM (United States).

- Fettig, A., & Lefkowitz, G. (2005). Twisted network programming essentials. " O'Reilly Media, Inc."
- Forouzan, Behrouz A. *Tcp/Ip Protocol Suite*. McGraw-Hill Higher Education, 2002.
- Hätönen, Seppo; Rao, Ashwin; Tarkoma, Sasu. (2021). Programmable Session Layer Multi-Connectivity. *Ieee Access*, 10: 5736-5752.
- Hattingh, C. (2020). Using Asyncio in Python: understanding Python's asynchronous programming features. " O'Reilly Media, Inc."
- Lubanovic, B. (2014). *Introducing Python: Modern Computing in Simple Packages*. " O'Reilly Media, Inc."
- Lubanovic, B. (2022). *Python: Comprendre les bases et maîtriser la programmation*. De Boeck Supérieur.
- Rohith, R., Moharir, M., & Shobha, G. (2018, December). SCAPY-A powerful interactive packet manipulation program. *In 2018 international conference on networking, embedded and wireless systems (ICNEWS)*, (pp. 1-5). IEEE.
- SALIHU, Egzon; BLAKAJ, Gentiana. (2021) Workplace Chat Application Using Socket Programming in Python.
- Sarker, M. F., & Washington, S. (2015). *Learning Python Network Programming*. Packt Publishing Ltd.
- Shminke, Boris. (2022). Python client for Isabelle server. arXiv preprint arXiv:2212.11173.
- Soni, M., & Rajput, B. S. (2021). Security and performance evaluations of QUIC protocol. *In Data Science and Intelligent Applications: Proceedings of ICDSIA, 2020* (pp. 457-462). Springer Singapore.
- Thaker, N., & Shukla, A. (2020). Python as multi paradigm programming language. *International Journal of Computer Applications*, vol. 177, no. 31, (pp. 38-42).