WhaleConnect: A General-Purpose, Cross-Platform Network Communication Application

Aidan Sun [1]

1 Independent Researcher, United States

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor: Open Journals

Reviewers:

- @openjournals

Submitted: 01 January 1970
Published: unpublished

Summary

WhaleConnect, a general-purpose and cross-platform network communication application, aims to overcome existing network communication challenges, such as data security, efficiency when handling parallel connections, and compatibility of communication between diverse technologies. It runs on Windows, macOS, and Linux, and it supports various communication protocols. Highly scalable communication is achieved through parallel processing and operating system features, and the application has an intuitive graphical user interface. Through these features, WhaleConnect is designed to foster the development of network-enabled systems and applications, such as the Internet of Things, an emerging and rapidly growing field that involves network communication and data transfer between smart devices.

Statement of need

Computer networking is a vital component of modern computing that allows devices such as computers to communicate and share information. Having a global market size growth of 22% in 2021 to hit $157.9 billion, one emerging paradigm employing computer networking is the Internet of Things (IoT). IoT combines energy-efficient microcontrollers with sensors and actuators to create interconnected smart devices, such as smart traffic signals and automatic home lighting systems (Elgazzar, 2022). However, some crucial challenges in the development of IoT-based systems are security (ensuring the integrity of user data), scalability (supporting a large number of connected devices without degrading the performance of the system), and interoperability (being able to support the exchange of information across different technologies and platforms) (Kumar et al., 2019). Many tools currently implement network communication for simple IoT projects, though none completely address these challenges.

This application, WhaleConnect, is an open source network communication tool that supports Windows, macOS, and Linux. It implements communication through TCP and UDP over the Internet Protocol and L2CAP and RFCOMM over Bluetooth, enabling interoperability with a wide range of IoT and wireless-enabled devices. It also supports the TLS protocol for reliable data security and encryption. Additionally, it can function as both a client and a server, allowing it to provide services and information to some devices and request them from others to offer a complete solution for control. Scalable parallel communication is enabled by leveraging resources provided by the operating system.

All functionality is exposed through a graphical user interface (GUI), offering a seamless and intuitive user experience, especially when managing multiple connections over various protocols and devices. The user interface also offers other useful functions that can aid in diagnostics, such as displaying timestamps, hex dumps, and logs of sent data. Overall, WhaleConnect addresses many key challenges in the field of IoT, promoting further research and development in this rapidly-growing field. WhaleConnect aims to be user-friendly and widely used by researchers, developers, hobbyists, and the industry.

The Journal of Open Source Software

Architecture

WhaleConnect uses multithreading in conjunction with high-performance kernel functions — IOCP (Ashcraft, 2022) on Windows, kqueue (Lemon, 2000) and IOBluetooth (Apple Inc., n.d.) on macOS, and io_uring via liburing (Axboe, n.d.) on Linux — to fully benefit from computer hardware and offer a high degree of scalability when managing multiple connections at once. In addition, it uses coroutines, introduced in the C++ 2020 standard (cppreference Contributors, 2024), to efficiently manage concurrent connections within each thread. Figure 1 presents an architecture diagram displaying the interactions between threads, the operating system's kernel, and the user interface.

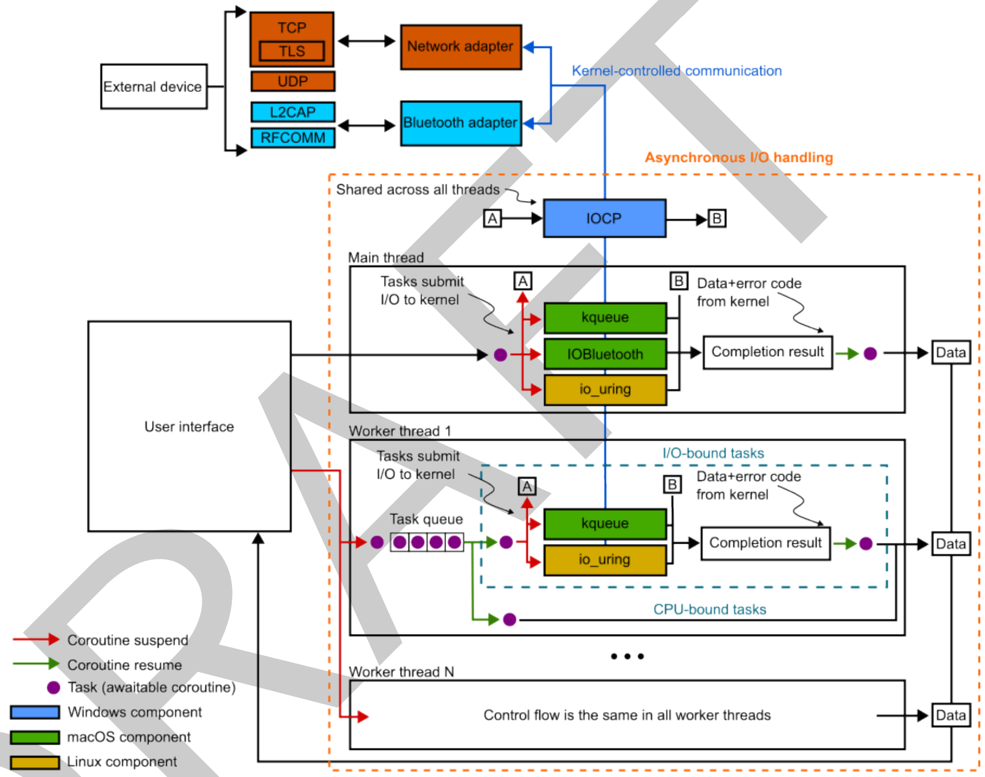

Figure 1: Architecture diagram of WhaleConnect

Acknowledgements

This research has received no external funding.

References

Apple Inc. (n.d.). *IOBluetooth*. https://developer.apple.com/documentation/iobluetooth

Ashcraft, A. et al. (2022). *I/O Completion Ports*. https://learn.microsoft.com/en-us/windows/win32/fileio/i-o-completion-ports

Axboe, J. (n.d.). *liburing library for io_uring*. https://github.com/axboe/liburing

cppreference Contributors. (2024). *Coroutines (C++20)*. https://en.cppreference.com/w/cpp/language/coroutines

Elgazzar, K. et al. (2022). Revisiting the internet of things: New trends, opportunities and grand challenges. *Frontiers in the Internet of Things*. https://doi.org/10.3389/friot.2022.1073780

Kumar, S., Tiwari, P., & Zymbler, M. (2019). Internet of things is a revolutionary approach for future technology enhancement: A Review - Journal of Big Data. *SpringerOpen*. https://doi.org/10.1186/s40537-019-0268-2

Lemon, J. (2000). *Mac OS X Manual Page For kqueue(2)*. https://developer.apple.com/library/archive/documentation/System/Conceptual/ManPages_iPhoneOS/man2/kqueue.2.html