Proposal for Enhancing Network Benchmark Tests

# Introduction

In exploring the network benchmarking tests currently available under `/network/benchmarks/netperf`, it has become evident that these tests are both outdated and restrictive. The process of adding new tests is convoluted, involving numerous conditionals that complicate the task. This proposal aims to address these issues by suggesting a cleaner, more flexible approach to writing network tests, along with incorporating additional output options such as JSON.

# Current Limitations

The present state of network benchmarking tests under `/network/benchmarks/netperf` suffers from several key limitations:

* The code structure is convoluted, making it difficult to add or modify tests.
* Output options are limited, reducing the utility and versatility of the tests.

# Proposed Enhancements

To overcome these limitations, the following enhancements are proposed:

## 1. Refactor Test Structure

The current test structure will be refactored to make it more modular and easier to understand. This includes:

* Simplify conditional statements to enhance code readability and maintainability which will be replacing numerous `if-else` and `switch` statements with a more declarative syntax.
* Implement a flexible framework to facilitate the addition of new tests. Ideally I would like to make it as simple as adding an item to a list.

## 2. Add JSON Output Option

To make the tests more useful and developer-friendly, an additional output option will be introduced. JSON output will enable:

* Rich data representation, capturing more detailed information from the tests.
* Easy integration with other tools and platforms for data analysis and visualization.
* Facilitated saving and publishing of test results in a semi-standardized format.

## 3. Rewrite Parsing Logics

The current parsing logics will be rewritten to incorporate error handling and produce standardized outputs. Additionally, `iperf` has an option to produce JSON results, which is much easier to parse compared to text output. This includes:

* Improved error handling to ensure robustness and reliability.
* Standardized output formats to facilitate easier interpretation and integration.
* Utilizing `iperf`'s JSON output option for more straightforward parsing.

# Benefits

Implementing these enhancements will yield several significant benefits:

* Improved usability: Developers will find it easier to add, modify, and understand network tests.
* Enhanced data representation: JSON output will provide a richer and more flexible way to handle test results.
* Increased reliability: Better error handling and standardized outputs will ensure more dependable results.

# Conclusion

In conclusion, the proposed enhancements to the network benchmarking tests under `/network/benchmarks/netperf` will make them more useful, easier to change, and more developer-friendly. By refactoring the test structure, adding JSON output options, and rewriting the parsing logics, we can ensure that these tests remain relevant and valuable tools for network performance analysis. This proposal seeks approval to proceed with the outlined changes, promising a more efficient and effective network benchmarking process for all developers involved.