Independent Submission
Request for Comments: 9564
Category: Informational
ISSN: 2070-1721

M. Blanchet
Viagenie

1 April 2024

Faster Than Light Speed Protocol (FLIP)

Abstract

The recent advances in artificial intelligence (AI) such as large language models enable the design of the Faster than LIght speed Protocol (FLIP) for Internet. FLIP provides a way to avoid congestion, enhance security, and deliver faster packets on the Internet by using AI to predict future packets at the receiving peer before they arrive. This document describes the protocol, its various encapsulations, and some operational considerations.

Status of This Memo

This document is not an Internet Standards Track specification; it is published for informational purposes.

This is a contribution to the RFC Series, independently of any other RFC stream. The RFC Editor has chosen to publish this document at its discretion and makes no statement about its value for implementation or deployment. Documents approved for publication by the RFC Editor are not candidates for any level of Internet Standard; see Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc9564.

Copyright Notice

Copyright (c) 2024 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (https://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document.

Table of Contents

1. Introduction

ChatGPT was introduced to the public on 30 November 2022 [CHATGPT]. Since then, large language models (LLMs) have been used for a large variety of applications. It demonstrates the powerful ability to generate precise output based on the input and based on the appropriate training of the LLM. This protocol specification uses this ability to predict future packets before they arrive at the receiving peer, therefore achieving faster-than-light-speed delivery, hence the protocol name: Faster than LIght speed Protocol (FLIP).

Since FLIP can predict packets, frames, strings, or byte streams, it could be used at any layer of the IP protocol stack. Moreover, with proper training, FLIP can also predict future encrypted packets, as encryption is just strings of bytes. This specification shows FLIP as a Layer 2 shim as well as a transport shim layer. Since FLIP can be used at any layer, it is expected that additional specifications will be created, such as predicting HTTP requests and answers, email content, and more.

Since communications in deep space are unfortunately limited to light speed, and given the very large distances between spacecrafts and Earth, the consequence is very long delays. By offering faster-than-light-speed delivery, FLIP is a key enabler and addition to deep-space IP networking [IP-DEEP-SPACE].

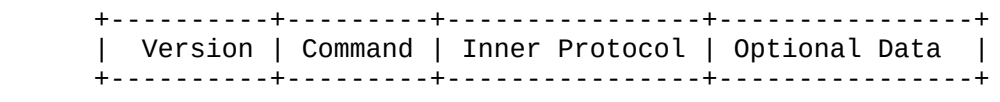
2. Protocol Peer Preparation

In order to successfully achieve faster than light speed, the peers of any protocol layer used by FLIP must prepare their side of the connection with the right model trained for the specific case. This document does not dictate any specific LLM, as the implementations may choose the one that best works for their use case and train them accordingly. As with any LLM, it is paramount to use a lot of training data, such as packet captures, in a variety of conditions to produce the best trained model. To avoid security, privacy, and legal issues, the specifics of which LLM is used, how it was trained, and what is the data set used, shall not be published nor disclosed in the protocol.

As an example, an implementation may elect to collect a significant number of Packet Capture (PCAP) files from tcpdump wiretapping at various vantage points on the Internet. The fact that traffic may be encrypted is not an issue, since a well-trained LLM will be able to predict encrypted traffic as accurately as unencrypted traffic.

3. FLIP Header

Wherever FLIP is used (below IP, above IP or other transport, or at the application layer), a FLIP shim header is inserted.



The header contains the following fields:

Version: A field of variable and unspecified length that contains the SHA-256 hash of the model, used as the version, as described in Section 5.

Command: The codepoint identifying the operation of this FLIP frame. Commands are described in Section 4. The initial list of valid FLIP commands is below.

The maximum number size is infinite, given that artificial intelligence peers can support an infinite number of commands, by just updating their models without the need to update their

protocol implementation.

Command	Codepoint	Reference
model	0x01	RFC 9564
data	0x02	RFC 9564

Table 1

Inner Protocol: As the FLIP header is a shim header, the inner protocol is specified in this field. For example, for a FLIP shim header inserted between IP and TCP, the IP packet will contain the FLIP codepoint as the transport protocol. The FLIP inner protocol field will then contain the TCP codepoint that would otherwise be in the IP packet.

Optional Data: Some commands have additional data that are following the Command field.

The header length is variable and depends on which command is used. Given the use of artificial intelligence by implementations of this protocol, the actual length of the header, and the length of each of its fields, is not specified in the header. Instead, it is expected that the proper neural network on the receiver side will be able to find the actual header termination, thus saving many header bits.

To properly signal the upper layer about the presence of the FLIP header, a specific codepoint is reserved at the layer below FLIP. Section 7 lists the registrations for IP and transport codepoints for this use.

4. Protocol Operation

Prior to sending a first packet using FLIP, the sender and the receiver should be configured with the appropriate model trained as discussed before. It is left to the implementation to choose the right LLM and the right training data set.

The following commands are defined:

Model: (codepoint 0x01). This command provides a way for peers to send their model in-band of the FLIP protocol. The model itself is carried in the Optional Data field of the FLIP header. Prior to the actual model data, a MIME header is inserted with the proper media type. If the media type for the model does not exist, it should be registered in the IANA Media Type registry.

Data: (codepoint 0x02). This command tells the receiving peer that the data that follows can be predicted and therefore achieves faster-than-light-speed performance.

Sending the model in-band to the other peer is an operation that should be done rarely, as models may be large in size. Moreover, it actually discloses the model for any wiretapping adversary. Implementors may consider using a post-quantum cryptographic algorithm that is also immune to AI prediction, therefore a post-Quantum-AI cryptographic algorithm.

5. Versioning

As described in [RFC6709], most protocols should be designed to enable future enhancements, such as providing a way to signal a new version of the protocol. In the case of FLIP, trained models will

always be enhanced by new training. A SHA-256 [RFC6234] hash of the trained model is used as a version number so each peer knows which FLIP version is being used. The SHA-256 hash is put in version field in the FLIP header as described previously. Given that new SHA-256 hashes are not sequential but fully random, replay attacks of future predictions are prevented.

6. Future Work

This new protocol may revolutionize how we design Internet protocols and how we use the Internet. For example, it is envisioned that this protocol may be used for video streaming, augmented reality, virtual reality, and post-quantum cryptography to name a few. By predicting the future packets, all these protocols and applications can benefit the use of FLIP.

7. IANA Considerations

For FLIP, codepoints could be registered in the following IANA registries.

* Protocol Numbers [IANA-PN]: 345, FLIP, Faster than LIght speed Protocol, RFC 9564

* Service Name and Transport Protocol Port Number Registry [IANA-SN]: FLIP, 68534, udp and tcp, RFC 9564

8. Security Considerations

The ability to predict future packets based on LLMs can be used by adversaries that are listening to the traffic via wiretapping. If they have access to the same model used by the destination peer, they could use it to predict the next packets and then initiate various attacks, including novel ones such as the "futureplay attack." Compared to the typical replay attack, this attack is where the adversary will predict future packets and then send them in advance to the destination. While it may not be obvious at this time, these novel attacks should be investigated before they become a problem. Therefore, further research in this field is suggested.

The ability for a peer to predict future packets enhances the overall security of the Internet because adversaries will not be able to inject bad packets in a connection, as the destination will be able to compare the received bad packet with the calculated prediction and therefore will easily identify and deny any bad packets.

9. Informative References

[CHATGPT] Wikipedia, "ChatGPT", 20 March 2024, <https://en.wikipedia.org/w/index.php?title=ChatGPT&oldid=1214732037>.

[IANA-PN] IANA, "Protocol Numbers", <https://www.iana.org/assignments/protocol-numbers/>.

[IANA-SN] IANA, "Service Name and Transport Protocol Port Number Registry", <https://www.iana.org/assignments/service-names-port-numbers/>.

[IP-DEEP-SPACE] Blanchet, M., Huitema, C., and D. Bogdanovi, "Revisiting the Use of the IP Protocol Stack in Deep Space: Assessment and Possible Solutions", Work in Progress, Internet-Draft, draft-many-deepspace-ip-assessment-01, 4 March 2024, <https://datatracker.ietf.org/doc/html/draft-many-deepspace-ip-assessment-01>.

[RFC6234] Eastlake 3rd, D. and T. Hansen, "US Secure Hash Algorithms (SHA and SHA-based HMAC and HKDF)", RFC 6234, DOI 10.17487/RFC6234, May 2011, <https://www.rfc-editor.org/info/rfc6234>.

[RFC6709] Carpenter, B., Aboba, B., Ed., and S. Cheshire, "Design Considerations for Protocol Extensions", RFC 6709, DOI 10.17487/RFC6709, September 2012, <https://www.rfc-editor.org/info/rfc6709>.

Acknowledgements

Since this protocol specification is using artificial intelligence and large language models, it was deemed that dumb humans must not review this specification. Instead, the specification has been submitted to multiple LLM chat services and was enhanced by their comments and suggestions, hence acknowledged here. In fact, this specification may have been produced entirely by LLM chat services. Moreover, given the specifications being produced by the IETF relying upon human intelligence, using LLMs to produce specifications should be envisioned. Finally, given the difficulty to find experts for management positions such as in the IESG or IAB, the use of LLMs should be considered to replace those roles. Unfortunately, given privacy, security, and legal considerations, the LLM chat services used for this specification cannot be named here.

Author’s Address

Marc Blanchet
Viagenie
Email: marc.blanchet@viagenie.ca