Transport and Services Working Group
Internet-Draft
Intended status: Informational
Expires: 8 January 2026

M. Seemann
Smallstep
M. Inden
Mozilla

7 July 2025

Controlling IP Fragmentation on Common Platforms
draft-seemann-tsvwg-udp-fragmentation-02

Abstract

When performing Path MTU Discovery (PMTUD) over UDP, applications must prevent fragmentation of UDP datagrams both by the sender's kernel and during network transit. This document provides guidance for implementers on configuring socket options to prevent fragmentation of IPv4 and IPv6 packets across commonly used platforms.

Discussion Venues

This note is to be removed before publishing as an RFC.

Discussion of this document takes place on the Transport and Services Working Group Working Group mailing list (tsvwg@ietf.org), which is archived at https://mailarchive.ietf.org/arch/browse/tsvwg/.

Source for this draft and an issue tracker can be found at https://github.com/marten-seemann/draft-seemann-tsvwg-udp-fragmentation.

Status of This Memo

Table of Contents

1. Introduction

[RFC0791] defines the Don't Fragment (DF) bit in the IPv4 header. When set, this bit prevents routers from fragmenting IP packets. If a router needs to fragment a packet with the DF bit set, it will instead drop the packet and send an ICMP "Fragmentation Needed" message back to the sender.

The DF bit has historically been most relevant to TCP ([RFC9293]), where the kernel handles Path MTU Discovery (PMTUD) internally. Applications using TCP sockets do not need to interact with the DF bit directly.

In IPv6 ([RFC8200]), fragmentation by intermediate nodes is not permitted. All IPv6 packets effectively have the DF bit set, however, the sender's kernel might still break up UDP datagrams that are too large to fit the MTU of the interface before sending a packet into the network.

[RFC8899] defines Datagram Packetization Layer Path MTU Discovery (DPLPMTUD), a mechanism that allows protocols running over UDP to determine the maximum packet size they can send. Setting the DF bit is crucial for DPLPMTUD, as it ensures that packets larger than the Path MTU are dropped, allowing the endpoint to detect the MTU limitation.

QUIC [RFC9000] is one such protocol that runs over UDP and makes use of DPLPMTUD. As QUIC implementations typically run in user space, they need to configure the DF bit on their UDP sockets to perform DPLPMTUD correctly.

This document provides guidance for implementers on how to set the DF bit on UDP sockets across different platforms.

2. Conventions and Definitions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Setting the DF bit

While routers don’t fragment IPv6 packets in transit, the sender’s kernel will still fragment UDP datagrams that are too large to fit the MTU of the interface before sending a packet into the network. Therefore, operating systems offer socket options to control the fragmentation behavior for IPv6 packets.

For user-space implementations of DPLPMTUD, applications need to set the DF bit on IPv4 sockets and prevent fragmentation on IPv6 sockets.

3.1. Linux

PMTUD behavior is controlled via the IP_MTU_DISCOVER and IPV6_MTU_DISCOVER socket options at the IPPROTO_IP and IPPROTO_IPV6 levels. There are two closely related values: For IPv4, both IP_PMTUDISC_DO and IP_PMTUDISC_PROBE enable setting the DF bit. For IPv6, both IPV6_PMTUDISC_DO and IPV6_PMTUDISC_PROBE prevent fragmentation by the sender.

The difference between the former and the latter is that the former instructs the kernel to process ICMP "Fragmentation Needed" messages, while the latter does not.

When the kernel processes an ICMP "Fragmentation Needed" message, it will prevent the transmission of larger datagrams. Applications that wish to only respond to fully encrypted end-to-end signals (for example QUIC ([RFC9000])) might therefore wish to use IP_PMTUDISC_PROBE and IPV6_PMTUDISC_PROBE.

For dual-stack sockets, both IPv4 and IPv6 socket options can be set independently.

3.2. Apple

For IPv4, Apple platforms use the socket option of level IPPROTO_IP with name IP_DONTFRAG with value 1. For IPv6, IPV6_DONTFRAG with value 1 is used for the IPPROTO_IPV6 level.

However, dual-stack sockets are handled differently: To open a dual-stack socket, an IPv6 socket needs to be opened and the IPV6_V6ONLY option needs to be set to 0. This enables the socket to send both IPv4 and IPv6 packets. IPv4 packets must be sent using an IPv4-mapped IPv6 address.

When using a dual-stack socket, it is only necessary (and possible) to set the IPV6_DONTFRAG socket option. This results in the DF bit being set when sending IPv4 packets, and prevents fragmentation of IPv6 packets. It is not possible to control the fragmentation behavior of IPv4 and IPv6 separately.

3.3. Windows

For IPv4, Windows uses the socket option of level IPPROTO_IP with name IP_DONTFRAGMENT with value 1. For IPv6, IPV6_DONTFRAG with value 1 is used for the IPPROTO_IPV6 level.

Similar to the Apple platforms, dual-stack sockets are IPv6 sockets with the IPV6_V6ONLY option set to 0. IPv4 packets must be sent using an IPv4-mapped IPv6 address. However, contrary to Apple platforms, the DF bit on IPv4 packets is controlled by the IP_DONTFRAGMENT socket option.

4. Security Considerations

TODO Security

5. IANA Considerations

This document has no IANA actions.

6. Normative References

[RFC0791] Postel, J., "Internet Protocol", STD 5, RFC 791, DOI 10.17487/RFC0791, September 1981, <https://www.rfc-editor.org/rfc/rfc791>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/rfc/rfc2119>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/rfc/rfc8174>.

[RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <https://www.rfc-editor.org/rfc/rfc8200>.

[RFC8899] Fairhurst, G., Jones, T., T端xen, M., R端ngeler, I., and T. V旦lker, "Packetization Layer Path MTU Discovery for Datagram Transports", RFC 8899, DOI 10.17487/RFC8899, September 2020, <https://www.rfc-editor.org/rfc/rfc8899>.

[RFC9000] Iyengar, J., Ed. and M. Thomson, Ed., "QUIC: A UDP-Based Multiplexed and Secure Transport", RFC 9000, DOI 10.17487/RFC9000, May 2021, <https://www.rfc-editor.org/rfc/rfc9000>.

[RFC9293] Eddy, W., Ed., "Transmission Control Protocol (TCP)", STD 7, RFC 9293, DOI 10.17487/RFC9293, August 2022, <https://www.rfc-editor.org/rfc/rfc9293>.

Acknowledgments

TODO acknowledge.

Authors' Addresses

Marten Seemann
Smallstep
Email: martenseemann@gmail.com

Max Inden
Mozilla
Email: mail@max-inden.de