

**NAME**

udplite - Lightweight User Datagram Protocol

**SYNOPSIS**

```
#include <sys/socket.h>
```

```
sockfd = socket(AF_INET, SOCK_DGRAM, IPPROTO_UDPLITE);
```

**DESCRIPTION**

This is an implementation of the Lightweight User Datagram Protocol (UDP-Lite), as described in RFC 3828.

UDP-Lite is an extension of UDP (RFC 768) to support variable-length checksums. This has advantages for some types of multimedia transport that may be able to make use of slightly damaged datagrams, rather than having them discarded by lower-layer protocols.

The variable-length checksum coverage is set via a [setsockopt\(2\)](#) option. If this option is not set, the only difference from UDP is in using a different IP protocol identifier (IANA number 136).

The UDP-Lite implementation is a full extension of [udp\(7\)](#)—that is, it shares the same API and API behavior, and in addition offers two socket options to control the checksum coverage.

**Address format**

UDP-Litev4 uses the *sockaddr\_in* address format described in [ip\(7\)](#). UDP-Litev6 uses the *sockaddr\_in6* address format described in [ipv6\(7\)](#).

**Socket options**

To set or get a UDP-Lite socket option, call [getsockopt\(2\)](#) to read or [setsockopt\(2\)](#) to write the option with the option level argument set to **IPPROTO\_UDPLITE**. In addition, all **IPPROTO\_UDP** socket options are valid on a UDP-Lite socket. See [udp\(7\)](#) for more information.

The following two options are specific to UDP-Lite.

**UDPLITE\_SEND\_CSCOV**

This option sets the sender checksum coverage and takes an *int* as argument, with a checksum coverage value in the range 0..2<sup>16</sup>-1.

A value of 0 means that the entire datagram is always covered. Values from 1-7 are illegal (RFC 3828, 3.1) and are rounded up to the minimum coverage of 8.

With regard to IPv6 jumbograms (RFC 2675), the UDP-Litev6 checksum coverage is limited to the first 2<sup>16</sup>-1 octets, as per RFC 3828, 3.5. Higher values are therefore silently truncated to 2<sup>16</sup>-1. If in doubt, the current coverage value can always be queried using [getsockopt\(2\)](#).

**UDPLITE\_RECV\_CSCOV**

This is the receiver-side analogue and uses the same argument format and value range as **UDPLITE\_SEND\_CSCOV**. This option is not required to enable traffic with partial checksum coverage. Its function is that of a traffic filter: when enabled, it instructs the kernel to drop all packets which have a coverage *less* than the specified coverage value.

When the value of **UDPLITE\_RECV\_CSCOV** exceeds the actual packet coverage, incoming packets are silently dropped, but may generate a warning message in the system log.

**ERRORS**

All errors documented for [udp\(7\)](#) may be returned. UDP-Lite does not add further errors.

**FILES**

*/proc/net/snmp* - basic UDP-Litev4 statistics counters.

*/proc/net/snmp6* - basic UDP-Litev6 statistics counters.

**VERSIONS**

UDP-Litev4/v6 first appeared in Linux 2.6.20.

**BUGS**

Where glibc support is missing, the following definitions are needed:

```
#define IPPROTO_UDPLITE 136
#define UDPLITE_SEND_CSCOV 10
#define UDPLITE_RECV_CSCOV 11
```

**SEE ALSO**

[ip\(7\)](#), [ipv6\(7\)](#), [socket\(7\)](#), [udp\(7\)](#)

RFC 3828 for the Lightweight User Datagram Protocol (UDP-Lite).

*Documentation/networking/udplite.txt* in the Linux kernel source tree

**COLOPHON**

This page is part of release 4.10 of the Linux *man-pages* project. A description of the project, information about reporting bugs, and the latest version of this page, can be found at <https://www.kernel.org/doc/man-pages/>.