TLS is a protocol that is built on top of UDP instead of TCP. It is designed to provide the same security guarantees as TLS but with lower latency. DTLS is intended for the delivery of application data that is authenticated and encrypted end-to-end. [TLS is intended to deliver a stream of data reliably and with authenticated encryption, end-to-end 1](https://security.stackexchange.com/questions/29172/what-changed-between-tls-and-dtls).

DTLS has some differences with TLS 1.2 (RFC 5246). Most of the TLS elements are reused with only the smallest differences. The context is that the client and the server want to send each other a lot of data as “datagrams”; they really both want to send a long sequence of bytes, with a defined order, but do not enjoy the luxury of TCP. TCP provides a reliable bidirectional tunnel for bytes, where all bytes eventually reach the receiver in the same order as what the sender used; TCP achieves that through a complex assembly of acknowledge messages, transmission timeouts, and reemissions. This allows TLS to simply assume that the data will go unscathed under normal conditions; in other words, TLS deems it sufficient to detect alterations since such alterations will occur only when under attack. On the other hand, DTLS works over datagrams which can be lost, duplicated, or received in the wrong order. [To cope with that, DTLS uses some extra mechanisms and some extra leniency 1](https://security.stackexchange.com/questions/29172/what-changed-between-tls-and-dtls).

DTLS falling back to TLS can happen due to several reasons such as:

* The client does not support DTLS.
* The server does not support DTLS.
* The client or server does not support any common cipher suites.
* The client or server does not support any common compression methods.
* [The client or server does not support any common extensions 1](https://security.stackexchange.com/questions/29172/what-changed-between-tls-and-dtls).

I hope this helps!