

Kernel space

UDP / TCP socket

Encapsulator Prepares outgoing packets to the socket	"Decapsulator" Parses packets from the socket
Encryption Encrypts data which is being sent to the remote host	Decryption Decrypts data from the remote host
Authentication Sends authentication data to remote host	Authenticaiion Receives auth-data from remote host
Configuration Sends configuration settings to a remote host	Configuration Receives a remote configuration and parses it
Compressor Compresses data going to the remote host	Decompressor Decompresses data from the remote host
Packet mangler Modifies outgoing packets	Packet mangler Modifies incoming packets
Routing Routes a package to a remote host	Routing Routes a package from the tunnel to either local TUN/TAP or another remote
Packet consumer Reads packets from the tun/tap device	Packet producer Writes packets to the tun/tap device

TUN/TAP device

Kernel space

Socket module

Can be exchanged to support other transports than the default UDP/TCP.

SSL module

Separate modules which supports OpenSSL, GNUTLS, NSS, etc

Authentication module

Separate modules for each authentication scheme, like SSL certificate, username/password, etc

Configuration handshake

Exchange and agree on configuration parameters over-the-wire. Must support todays variant where both sides must have the same options, or a centralised model where server can override params.

Compression module

Modules supporting different compression algorithms

Protocol module

Modules supporting different network protocols, layer independent. This will cover IPv4, IPv6, SCCP, IGMP, etc. Mesh networking would be implemented at this level as well.

The purpose is make sure the packets have the correct attributes set and to route them to their proper destination, no matter if it is going to/from the local TUN/TAP device or being directed from a remote host to another remote host.

Consumer/producer module

Reads and writes data from the packet bus to/from the TUN/TAP device