


An introduction to smoltcp



Thibaut Vandervelden

What is smoltcp?

What is smoltcp?

- Open-source (<https://github.com/smoltcp-rs/smoltcp>)
- lightweight
- efficient
- TCP/IP stack implementation
- (LwIP, TinyTCP, uIP, ...)
- written in Rust 

Features of smoltcp

- Support for many protocols
 - UDP, TCP, IPv4, IPv6, IGMP, ICMPv4, ICMPv6, NDISC, ...
 - Parsing of Ethernet frames  and IEEE802.15.4 frames 
 - 6LoWPAN
- Modular and extensible architecture
- Low memory footprint

4 main components of smoltcp

The Device abstraction

-  Interaction with network device.

```
1  pub trait Device {
2      type RxToken: RxToken;
3      type TxToken: TxToken;
4
5      fn capabilities(&self) -> DeviceCapabilities;
6
7      fn receive(&mut self, timestamp: Instant) -> Option<(Self::RxToken, Self::TxToken)>;
8
9      fn transmit(&mut self, timestamp: Instant) -> Option<Self::TxToken>;
10 }
11
12 pub trait RxToken {
13     fn consume(self, f: impl FnOnce(&mut [u8]));
14 }
15
16 pub trait TxToken {
17     fn consume(self, len: usize, f: impl FnOnce(&mut [u8]));
18 }
```



The Interface

- 🚫 Filter incoming frames/packets
- ↕ Handle control messages
- 🔍 Provides lookup and caching of hardware addresses
- ↕ Dispatch packets to and from sockets

```
iface.poll(Instant::now(), &mut device, &mut sockets);
```

```
iface.poll_at(Instant::now(), &mut sockets);
```

The Sockets

-  Network endpoints and buffering
-  Protocol state machines
- Different from Berkeley socket interface
 - Buffer sizes are provided by programmer
- Interface-agnostic

```
iface.poll(Instant::now(), &mut device, &mut sockets);
```

- Supported sockets: ICMP, UDP, TCP, DHCPv4, DNS (client)

The Packets and Representations

Packets

- Zero-cost abstraction of frames and IP packets, enabling zero-copy reading

```
1 let frame = EthernetFrame::new_checked(&frame[..])?;  
2 let dst = frame.dst_addr();  
3  
4 let payload = frame.payload();
```

Representations

- Parsing packets for better validation

```
1 let repr = Icmpv6::parse(&frame)?;
```

- Emitting into packets

```
1 let buffer = [0;1500];  
2 let len = repr.buffer_len();  
3 repr.emit(  
4     &mut Icmpv6Packet::new_unchecked(&mut buffer[..len])  
5 );
```


Drawbacks of using smoltcp

- Unstable API (not yet a major version release)
- Limited protocol support compared to LwIP, uIP, TinyTCP, ...
- No bindings for C

Benefits of using smoltcp


- Written in Rust 
- Fast (multiple GB/s on Linux machine)

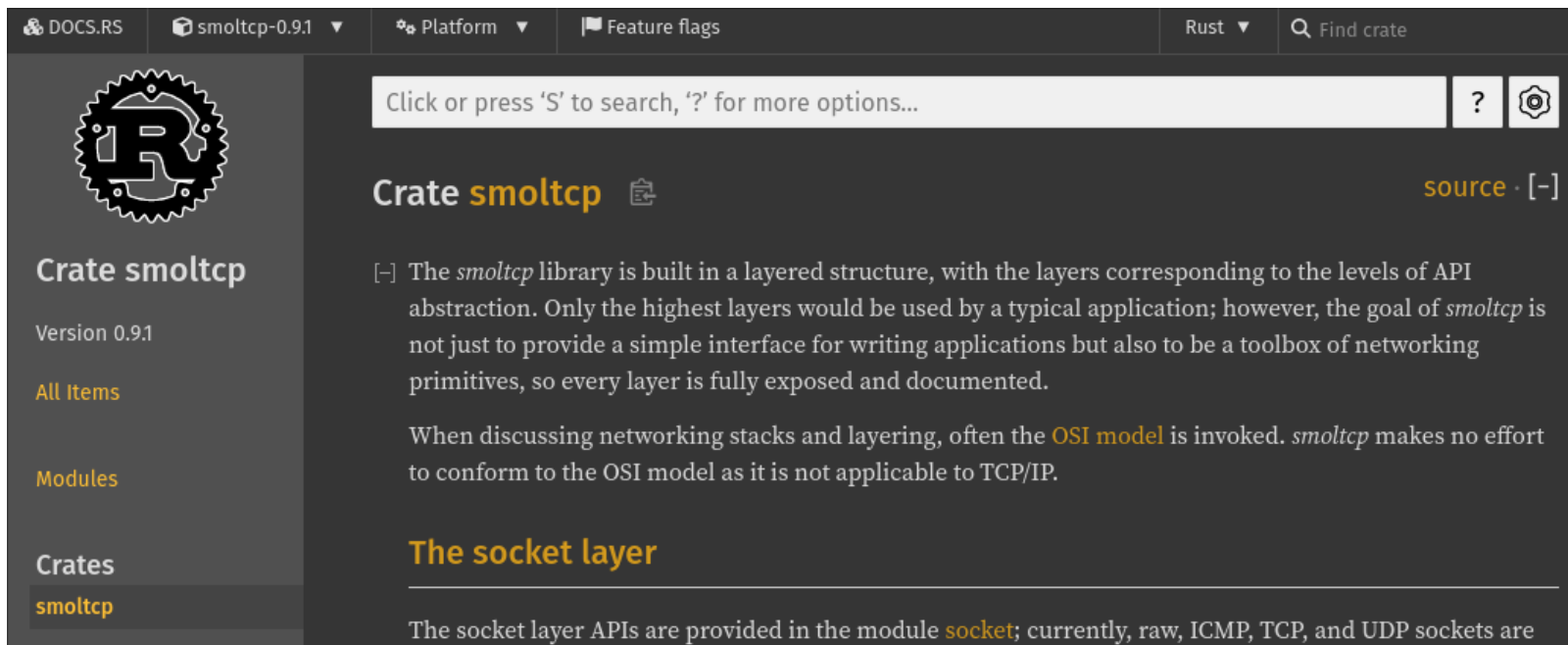
Benefits of using smoltcp

- Written in Rust 
- Fast (multiple GB/s on Linux machine)
- Highly configurable

```
1  [dependencies.smoltcp]
2  version = "0.9"
3  default-features = false
4  features = [
5      "medium-ieee802154",
6      "proto-sixlowpan", # Enables proto-ipv6
7      "proto-sixlowpan-fragmentation",
8      "socket-udp",
9  ]
```

Benefits of using smoltcp

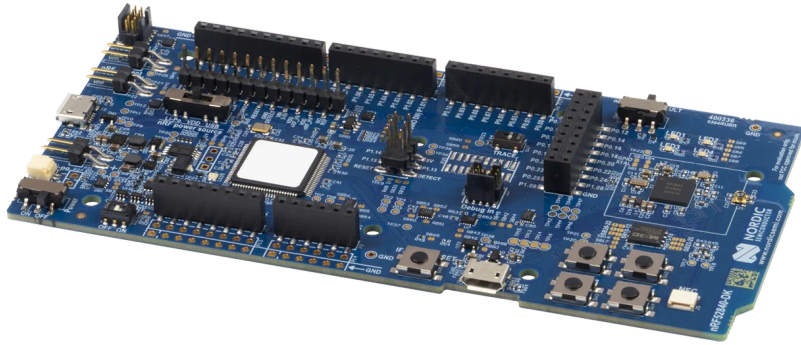
- Written in Rust 
- Fast (multiple GB/s on Linux machine)
- Highly configurable
- Very well documented (<https://docs.rs/smoltcp/latest/smoltcp/>)



The screenshot shows the documentation page for the `smoltcp` crate on `docs.rs`. The top navigation bar includes links for `DOCS.RS`, the crate version `smoltcp-0.9.1`, a `Platform` dropdown, `Feature flags`, the language `Rust`, and a search bar labeled `Find crate`. The left sidebar features the Rust logo, the crate name `Crate smoltcp`, the version `Version 0.9.1`, and links for `All Items`, `Modules`, and `Crates` (with `smoltcp` highlighted). The main content area has a search bar with the text `Click or press 'S' to search, '?' for more options...`. Below the search bar, the crate name `Crate smoltcp` is displayed with a clipboard icon and a `source` link. The description states: `[-] The smoltcp library is built in a layered structure, with the layers corresponding to the levels of API abstraction. Only the highest layers would be used by a typical application; however, the goal of smoltcp is not just to provide a simple interface for writing applications but also to be a toolbox of networking primitives, so every layer is fully exposed and documented.` It then explains that when discussing networking stacks, the `OSI model` is often invoked, but `smoltcp` does not conform to it as it is not applicable to TCP/IP. A section titled `The socket layer` follows, stating that the socket layer APIs are provided in the `socket` module, including raw, ICMP, TCP, and UDP sockets.

Use of smoltcp

- Embassy uses smoltcp as their network stack
- We used it on the nRF52840 and Zolertia RE-Mote



Stats Overview

↓ **345,171**

Downloads all time



