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.

```
pub trait Device {
    type RxToken: RxToken;
    type TxToken: TxToken;

    fn capabilities(&self) -> DeviceCapabilities;

    fn receive(&mut self, timestamp: Instant) -> Option<(Self::RxToken, Self::TxToken)>;

    fn transmit(&mut self, timestamp: Instant) -> Option<Self::TxToken>;
}

pub trait RxToken {
    fn consume(self, f: impl FnOnce(&mut [u8]));
}

pub trait TxToken {
    fn consume(self, len: usize, f: impl FnOnce(&mut [u8]));
}
```

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

```
let frame = EthernetFrame::new_checked(&frame[..])?;
let dst = frame.dst_addr();

let payload = frame.payload();
```

Representations

- Parsing packets for better validation

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

- Emitting into packets

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

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)

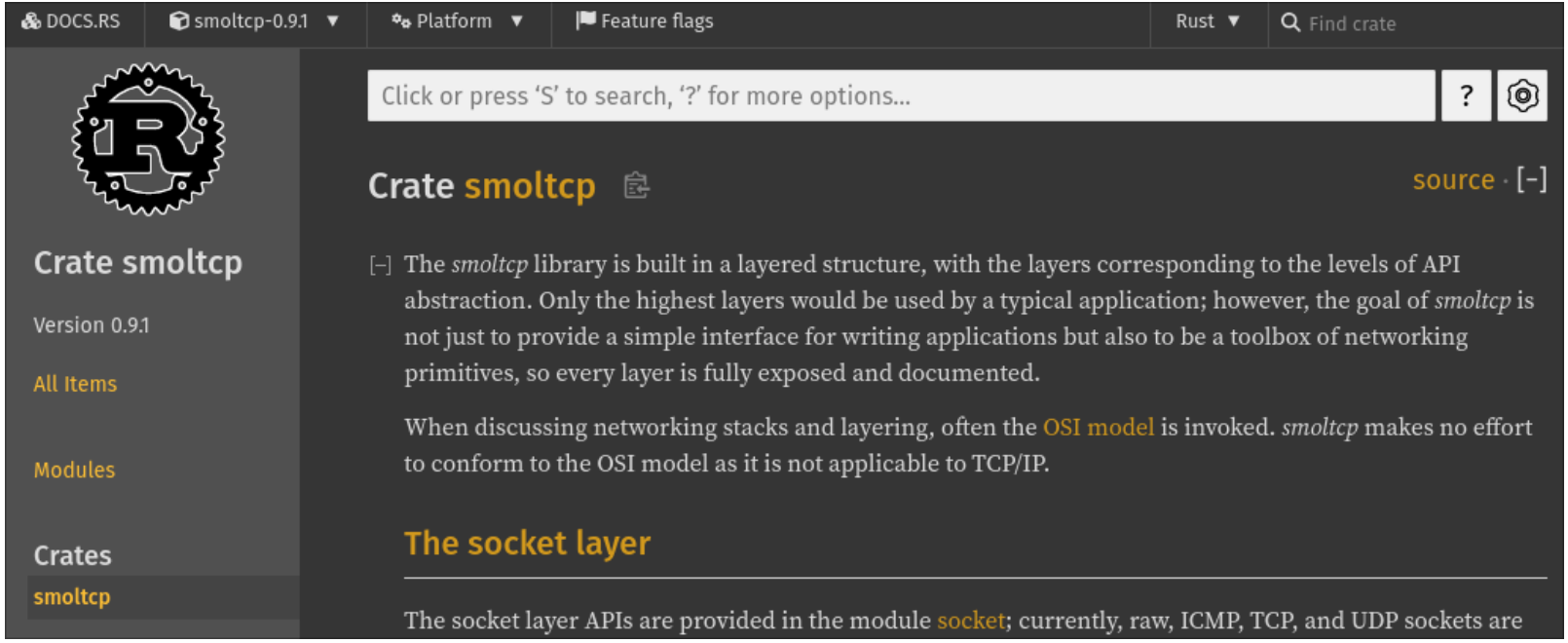
Benefits of using smoltcp

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

```
[dependencies.smoltcp]
version = "0.9"
default-features = false
features = [
    "medium-ieee802154",
    "proto-sixlowpan", # Enables proto-ipv6
    "proto-sixlowpan-fragmentation",
    "socket-udp",
]
```

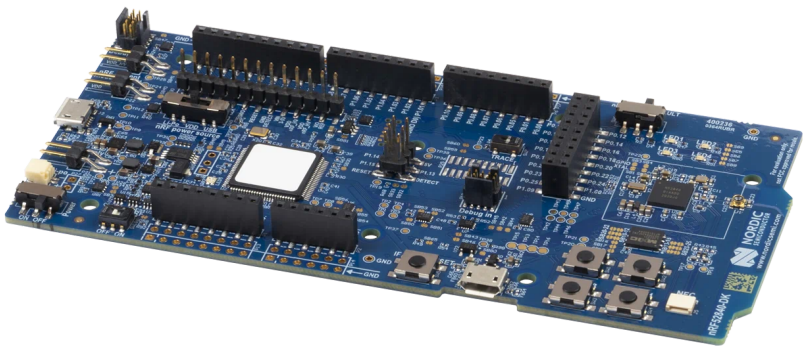
Benefits of using smoltcp

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



Use of smoltcp

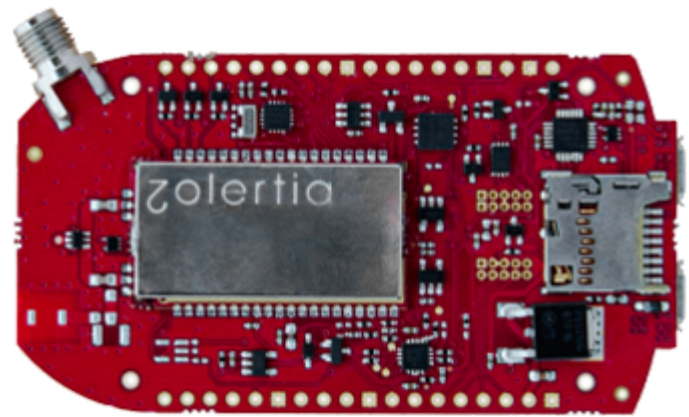
- Embassy uses smoltcp as their network stack
- We used it on the nRF52840 and Zolertia RE-Mote



Stats Overview

345,171

Downloads all time



END