

Verification of a UDP implementation

Daniel Neshyba-Rowe,
Ellen Whalen, Alain Kägi,
Wade McDermott, Nadiya Suleska



Lewis & Clark College

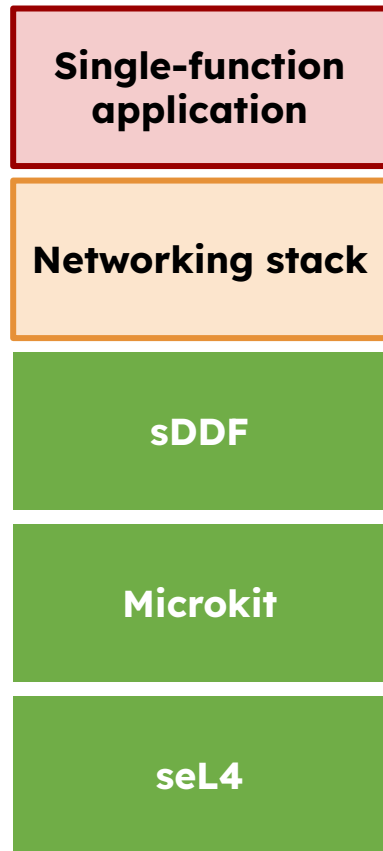


The Vision

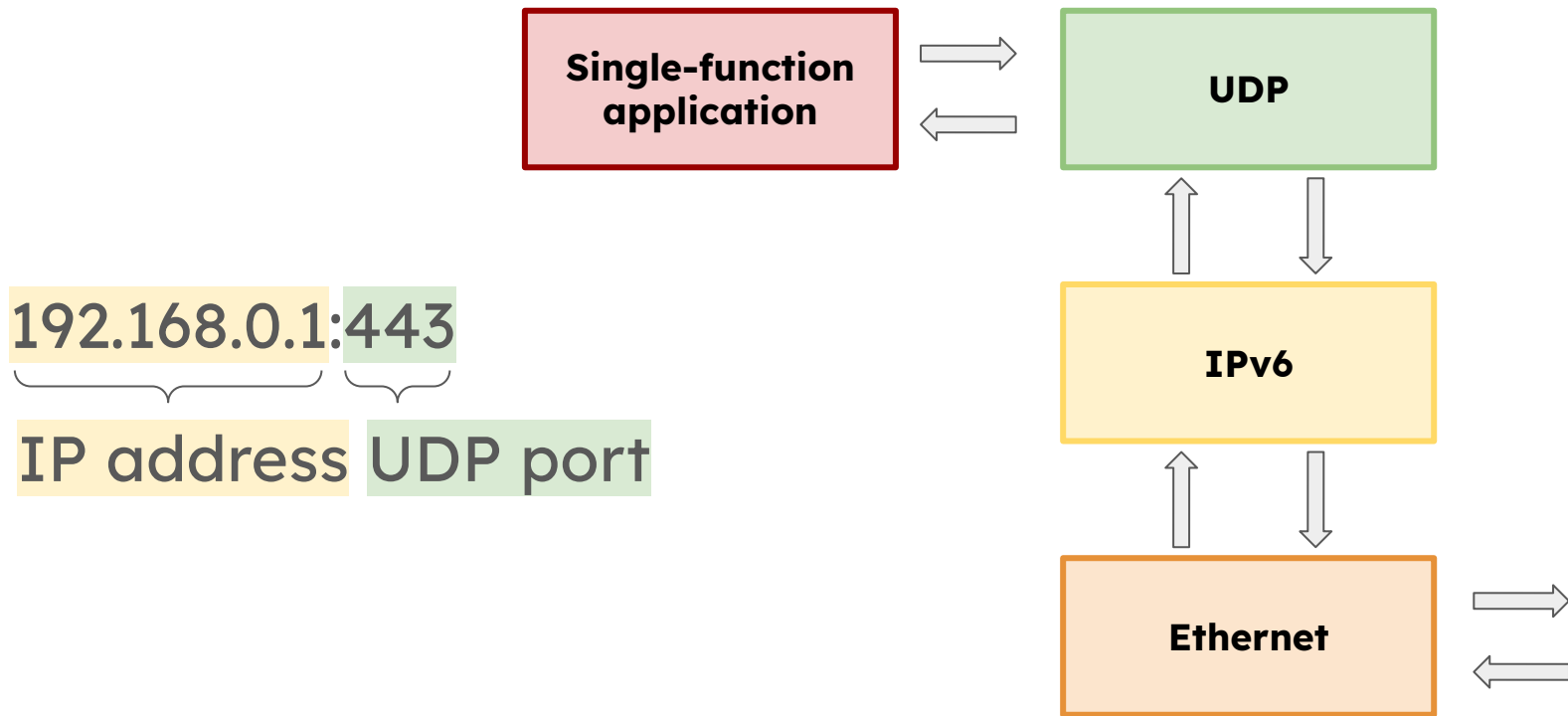
- Single-function, networked device
- Fast
- Provable (by a team of undergraduates)

How do we get there?

- IPv6 network stack
 - Geared towards end-points
 - High-performance
 - Drop-in replacement for lwIP
- AutoCorres2



Networking stack

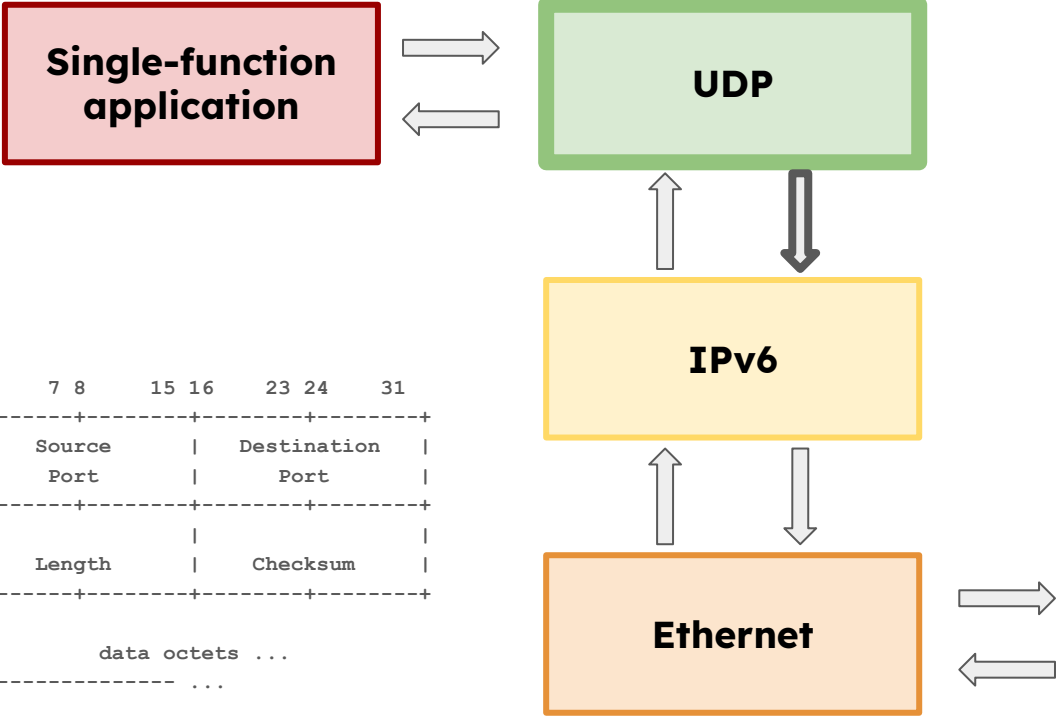


Networking stack

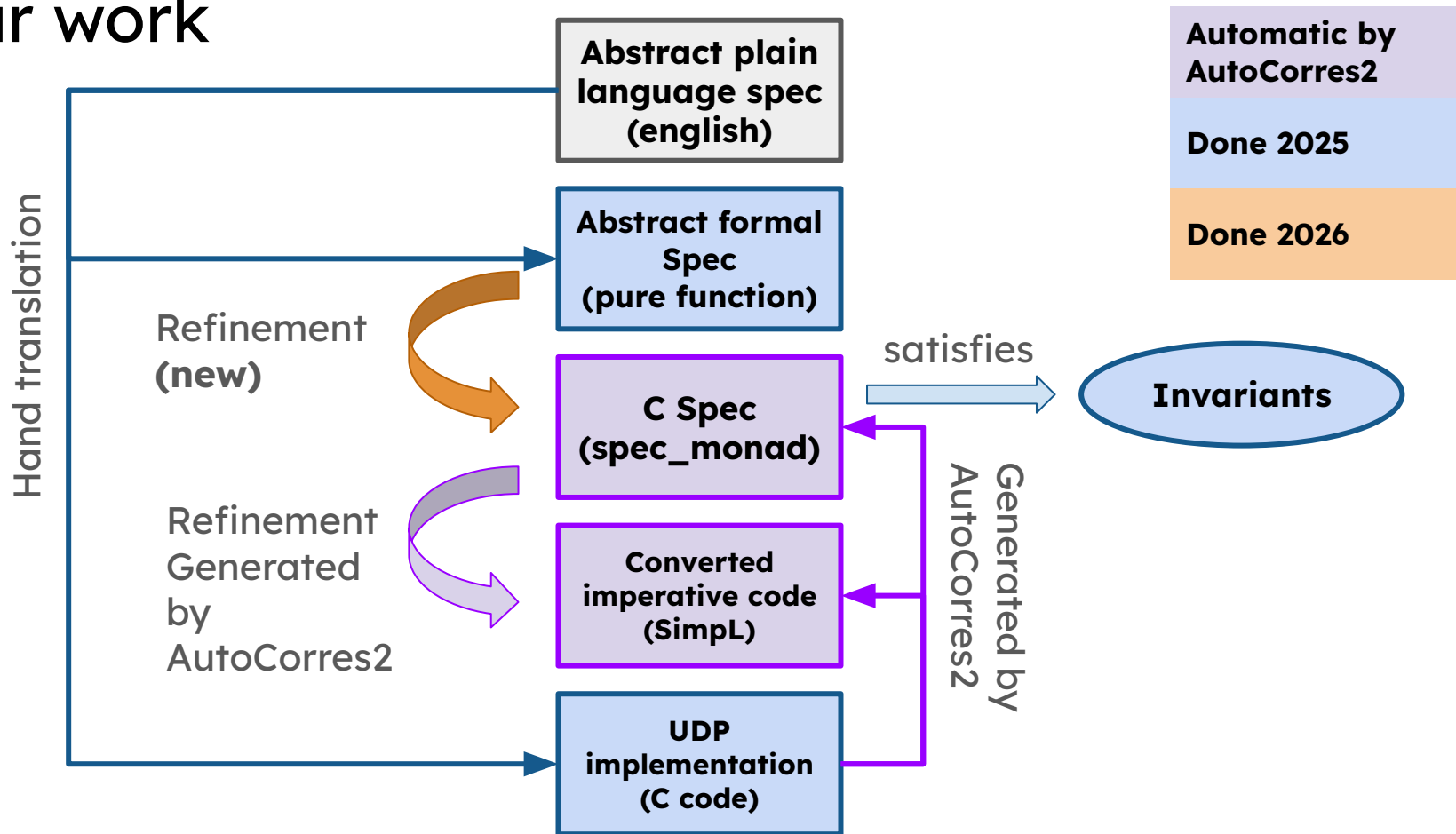
UDP wrap



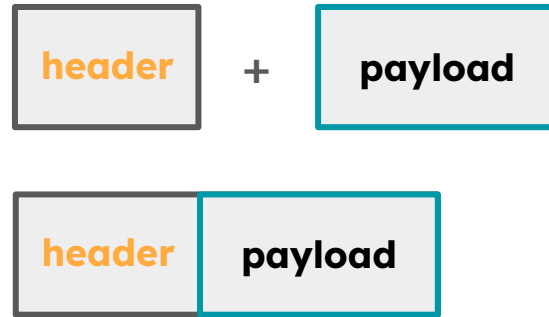
0	7	8	15	16	23	24	31
+-----+-----+-----+-----+							
Source				Destination			
Port				Port			
+-----+-----+-----+-----+							
Length				Checksum			
+-----+-----+-----+-----+							
data octets ...							
+-----+-----+-----+-----+							
...							



Our work

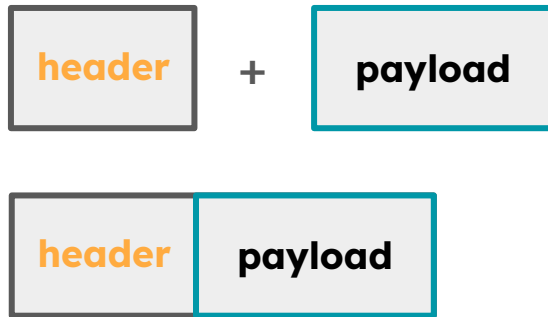


Abstract UDP wrap



```
definition udp_wrap :: <connection  $\Rightarrow$  octet list  $\Rightarrow$  octet list>  
  where <udp_wrap conn msg = (  
    let header = udp_wrap_header conn msg  
    in header @ msg)>
```

Abstract UDP wrap

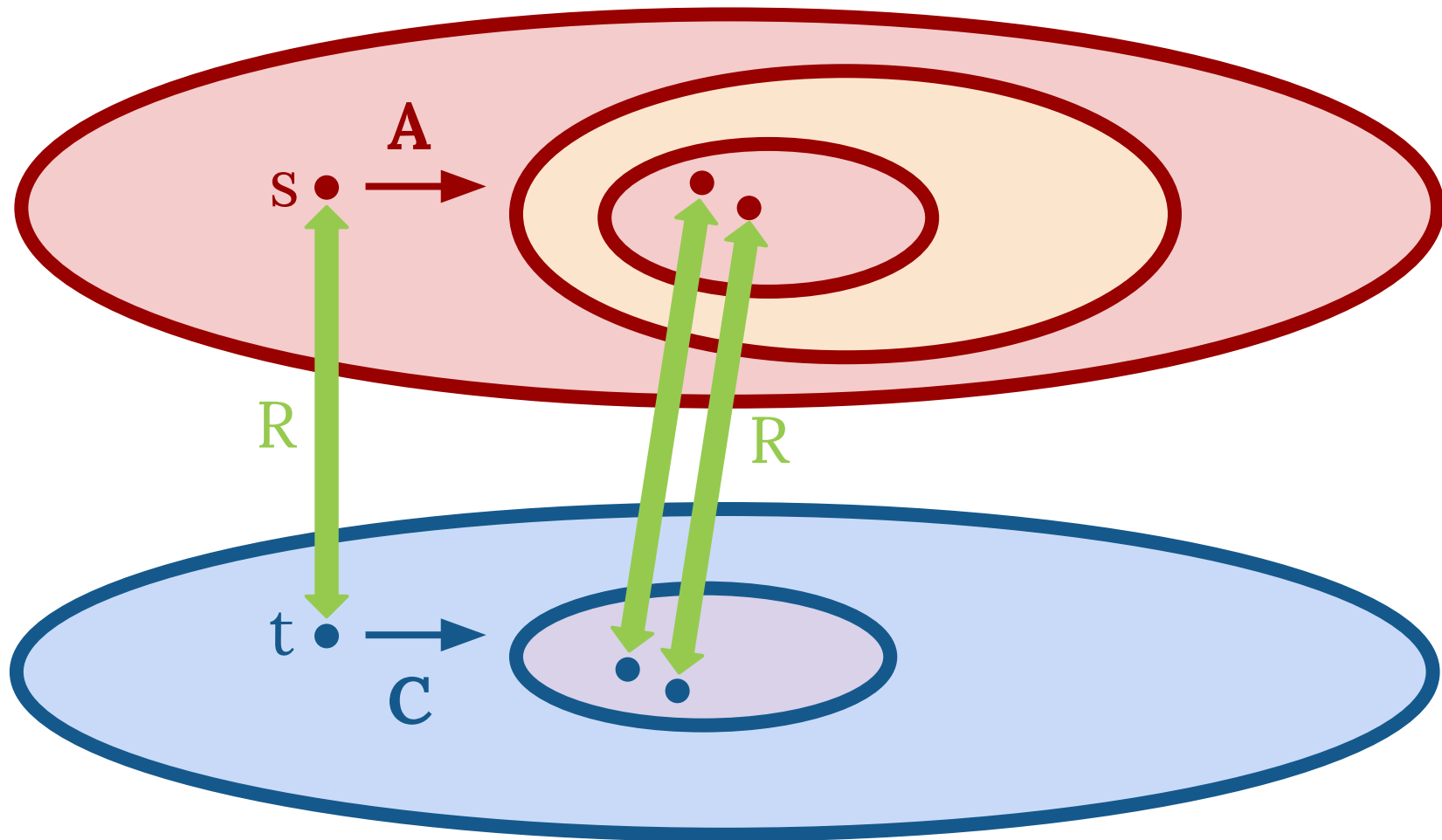


```
definition udp_wrap_buf :: <sk_buf  $\Rightarrow$  (unit, sk_buf, 's) spec_monad>  
  where <udp_wrap_buf sk  $\equiv$   
    do {  
      msg       $\leftarrow$  return (read_buf sk);  
      udp_header  $\leftarrow$  return (udp_wrap_header (connect sk) msg);  
      new_sk     $\leftarrow$  return (sk  $\sqcup$  data:=udp_header @ msg  $\sqcup$ );  
      return new_sk  
    }>
```

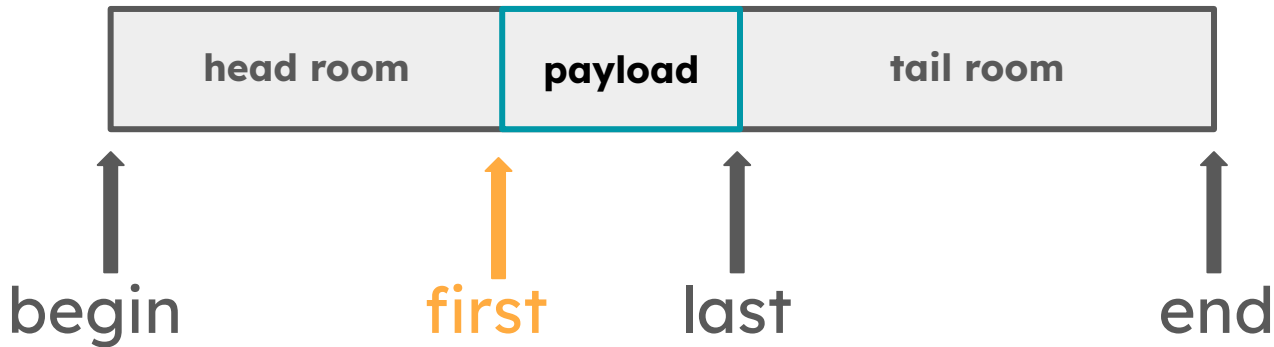
Refinement

C refines A when for any equivalent starting states, the possible resulting states from the execution of C is a ‘subset’ of the possible resulting states from the execution of A

```
lift_definition refines ::  
  <('e::default, 'a, 's) spec_monad =>  
    ('f::default, 'b, 't) spec_monad =>  
    's =>  
    't =>  
    (((('e, 'a) exception_or_result × 's) =>  
      (('f, 'b) exception_or_result × 't) => bool) =>  
      bool)>  
is <λf g s t R. sim_post_state R (f s) (g t)> .
```



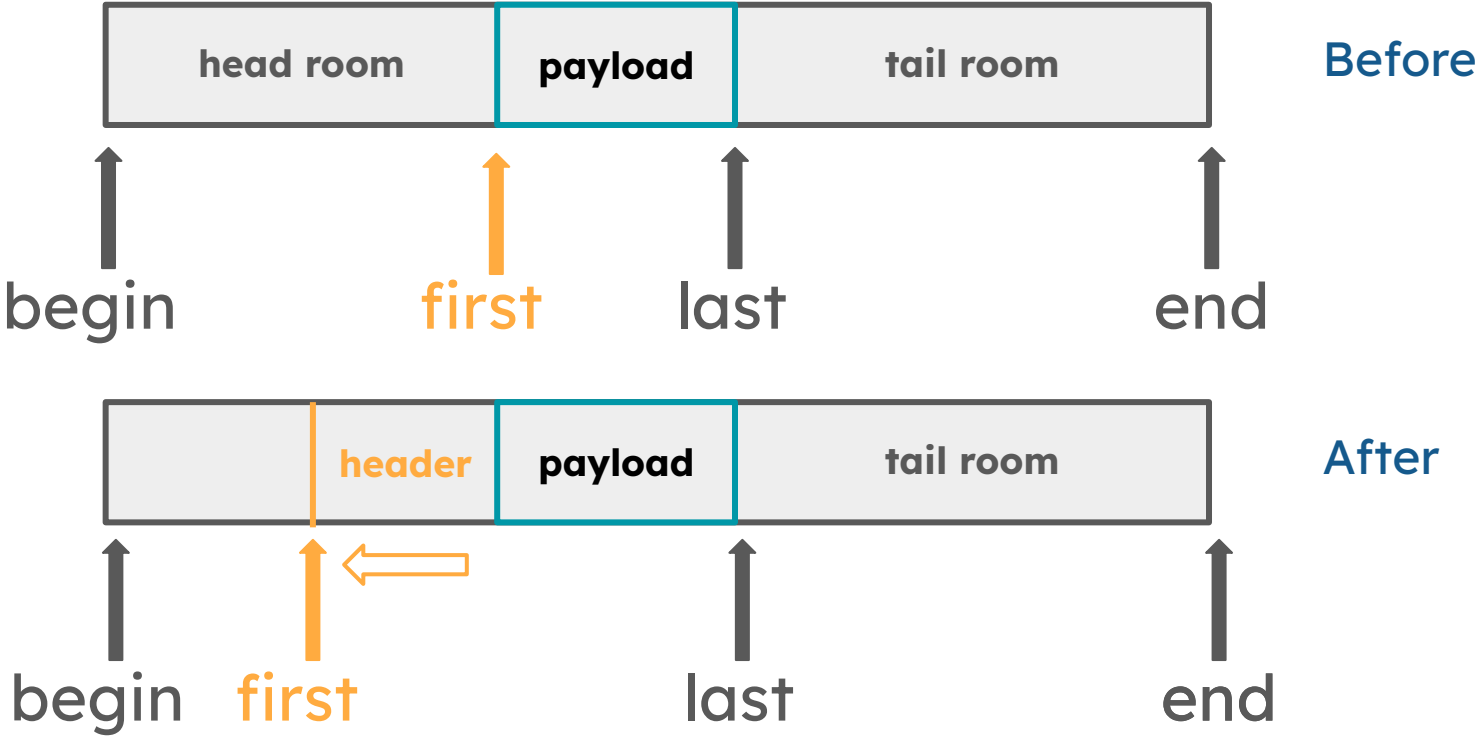
C UDP wrap (featuring Socket Buffer)



```
struct sk_buf {  
    uint8 *begin;    ///< The address of the first byte in the buffer  
    uint8 *end;      ///< The address of the first byte beyond the buffer  
    uint8 *first;    ///< The address of the first byte in the message  
    uint8 *last;     ///< The address of the first byte beyond the message  
};
```

Socket Buffer

C spec



Refinement for our purposes



Abstract spec



C spec

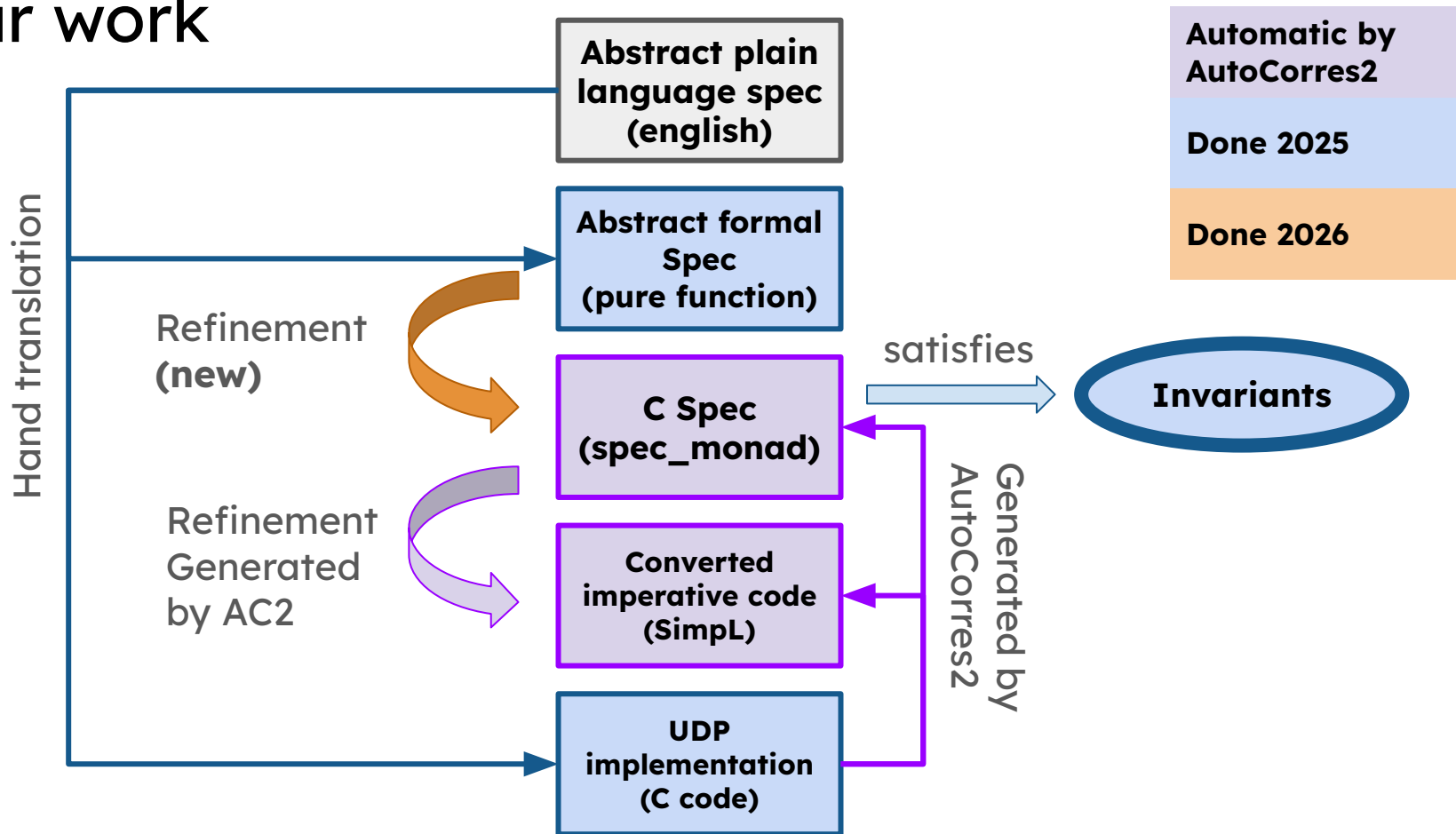
↑
first

↑
first

Before

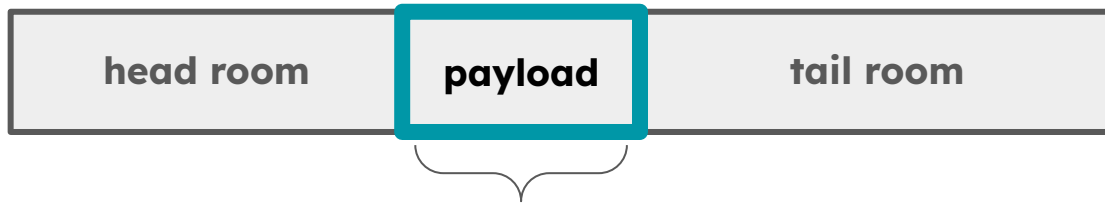
After

Our work



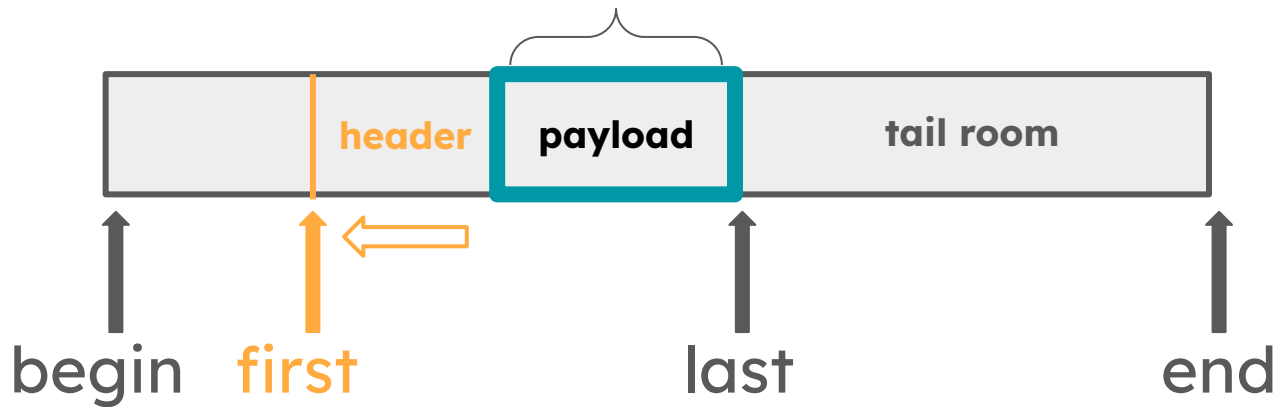
CSpec-level Invariants

Concrete



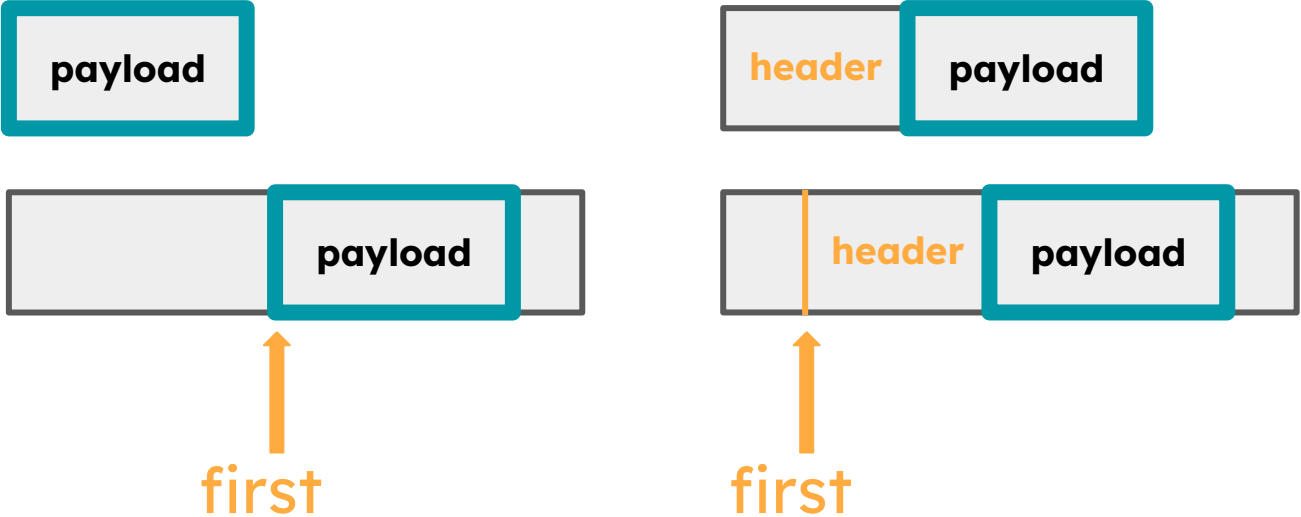
Before

||



After

Refinement for our purposes



Abstract

Concrete

Before

After

Refinement for our purposes



Abstract



Concrete

↑
first

↑
first

Before

After

Refinement proof snippet

“The 1st byte of the abstract and concrete headers agree after calling UDP wrap”

```
deref t' (first_C (deref t' sk_buf_c) +p 1) =  
snd (w16_to_octet (snd (fst (connect sk_buf_a))))
```

subgoal

```
apply (rule trans [where s=<fst (w16_to_octet (hton16_a (port_C (deref t (src_C (deref t sk_buf_c))))))>]])  
  apply (erule subst [where s=<src_port_C (deref t' (PTR_COERCE(8 word → udp_header_C) (first_C (deref t' sk_buf_c))))>]])  
  apply (subst w16_to_octet_def, subst word_split_def, simp)  
  apply (subst (1 3) h_val_def, subst ptr_val_ptr_coerce, subst (1 2) heap_list_def, simp)  
  apply (subst from_bytes_eq, subst udp_header_C_from_bytes, rule bit_word_eqI, simp add: ptr_add_def)  
apply (simp add: hton16_a_flips conn_sk_buf_equiv_def socket_equiv_def Let_def)  
done
```

Conclusions

- Fast — metrics looking promising (comparable to lwIP)
- Proofs — in progress, UDP wrap refinement done
- Undergraduates — steep learning curve, but doable
 - More documentation would be better
 - Accessible examples (Microkit is incredible)

Acknowledgements

- John S. Rogers Science Program
- Margaret Salstrom, Amy Timmins
- Dylan Angel, Aubrey Birdwell, Linus Brogan, Christian Ermann, Michael Harper, Seth Leichsenring, Jorge Martinez, Levi Overcast, Sydney Parker, Wyeth Greenlaw Rollins, Caitlyn Wilde, Natalie Zoz
- Johannes Åman Pohjola, June Andronick, Matthew Brecknell, Gernot Heiser, Gerwin Klein, Lucy Parker, Ivan Velickovic
- Jens Mache, Richard Weiss