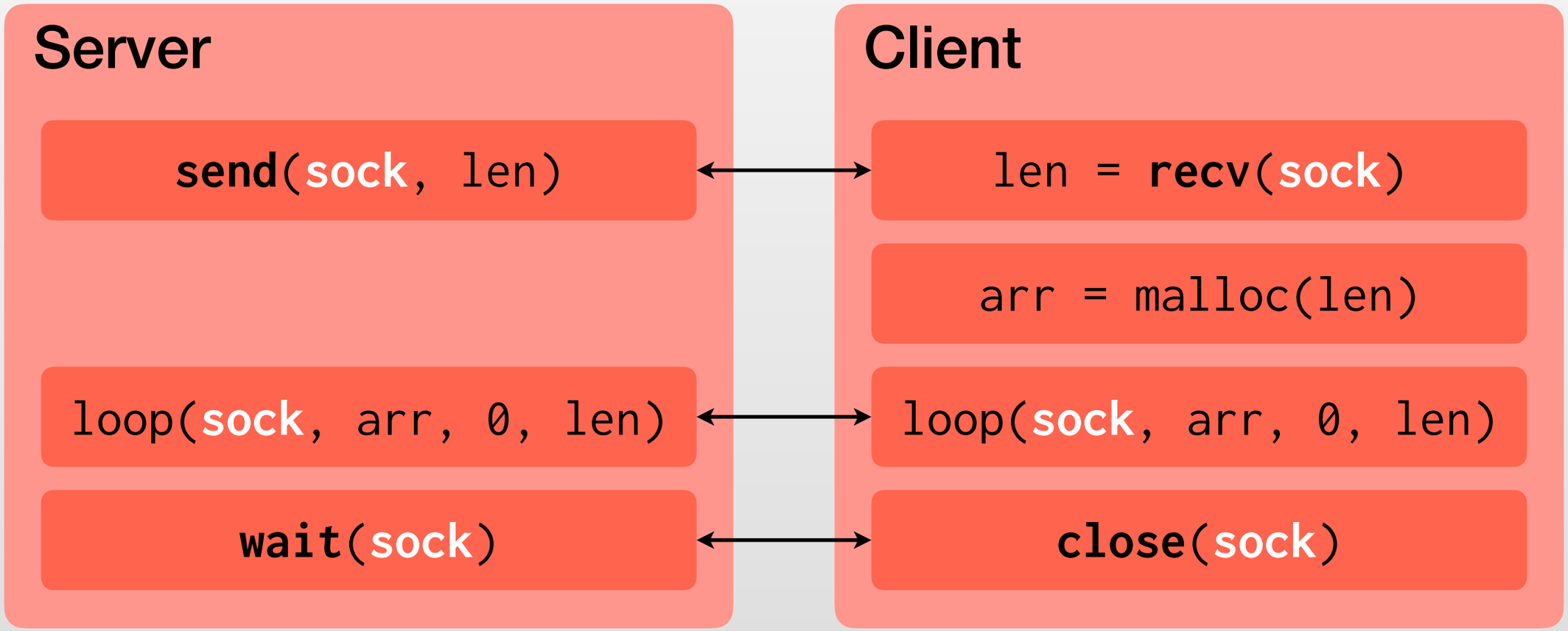
Dependent Session Types

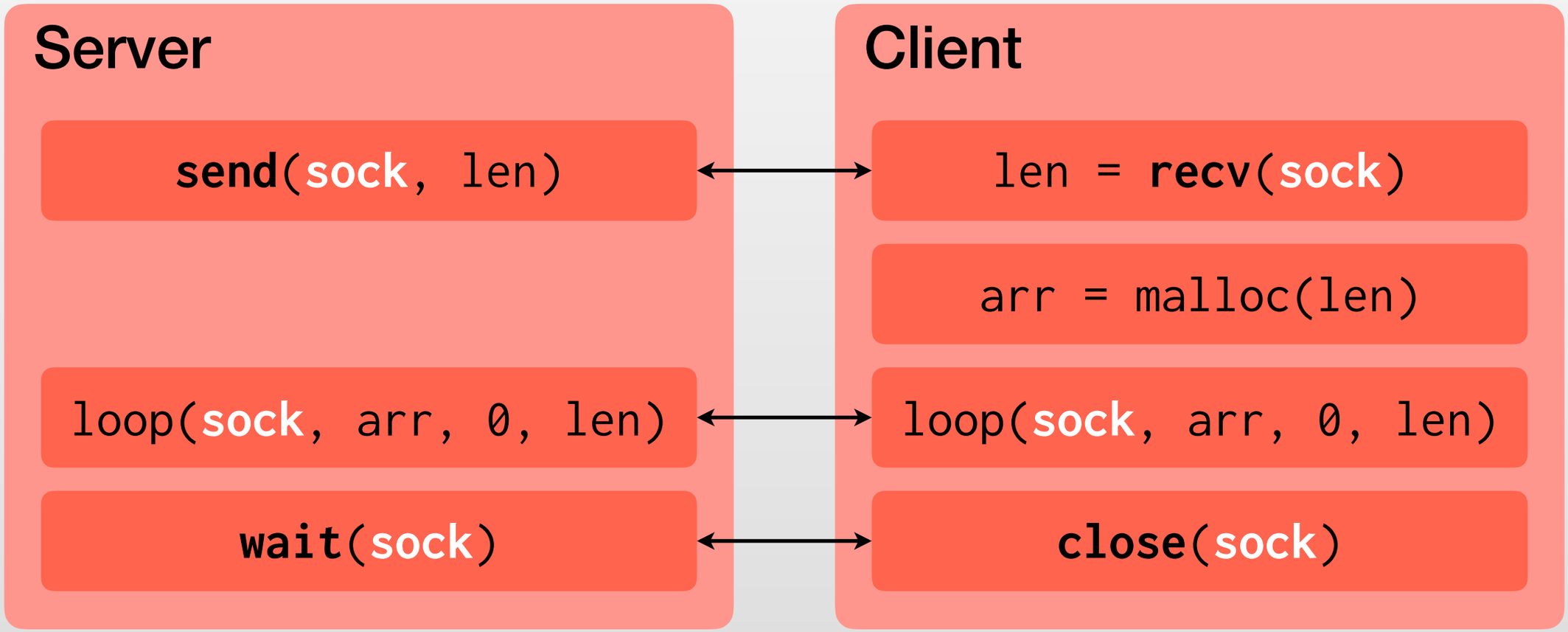
Hanwen Wu and Hongwei Xi
Boston University

IBM PL Day 2017 & NE/NJ PLS 2017

Sending an Array

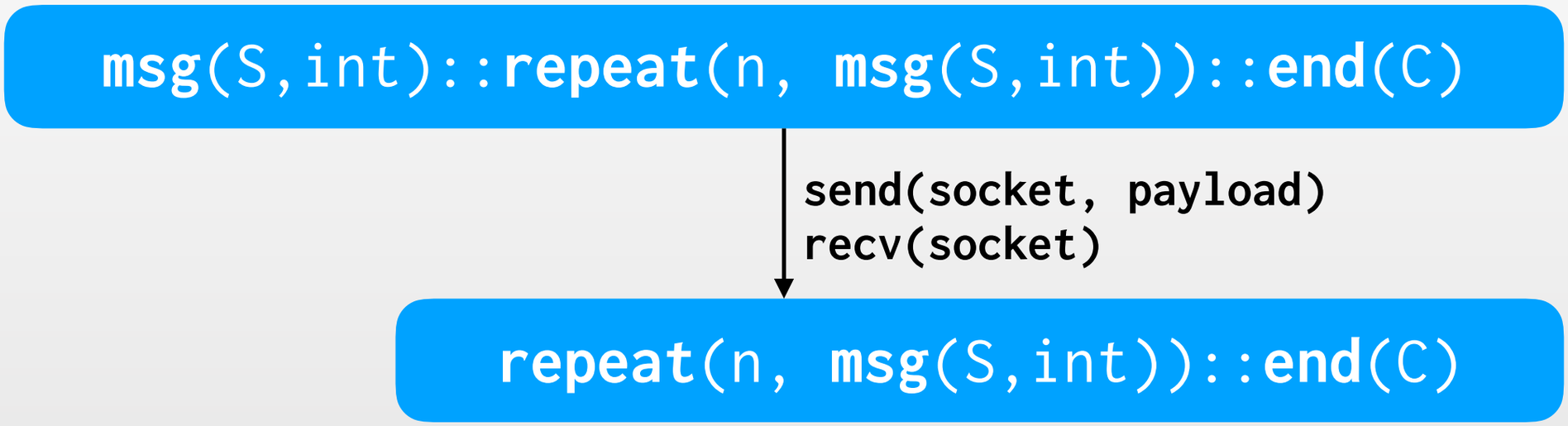


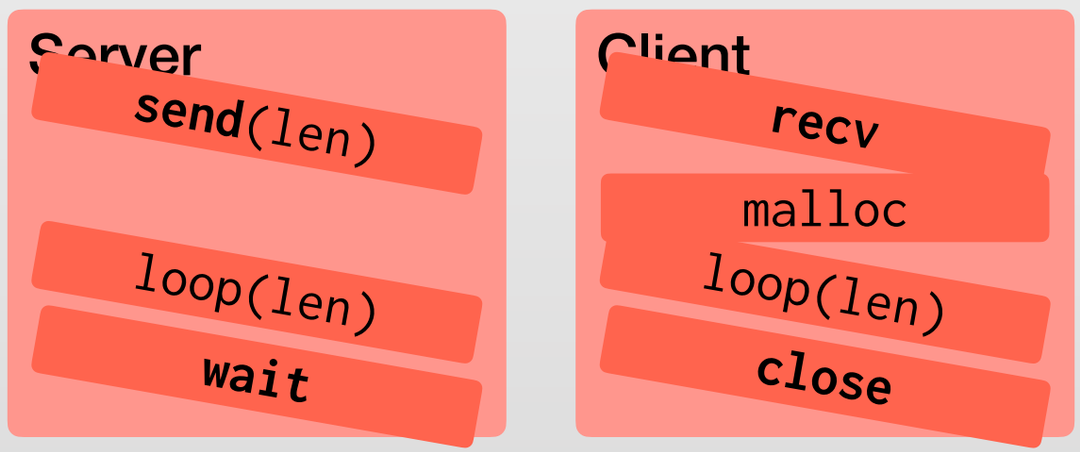
Sending an Array, Untyped



The socket is untyped.
What could go wrong?

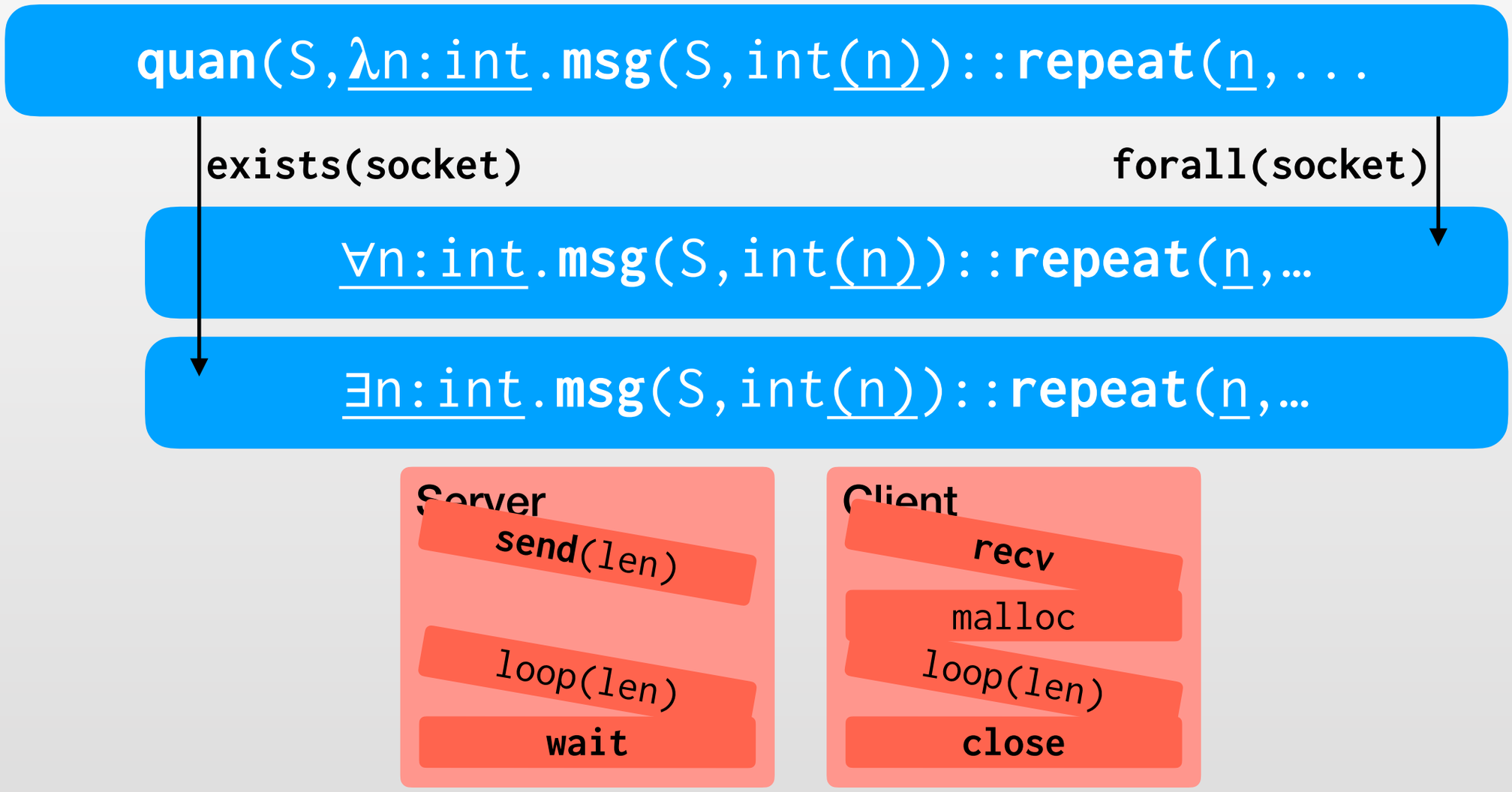
Sending an Array, Simple





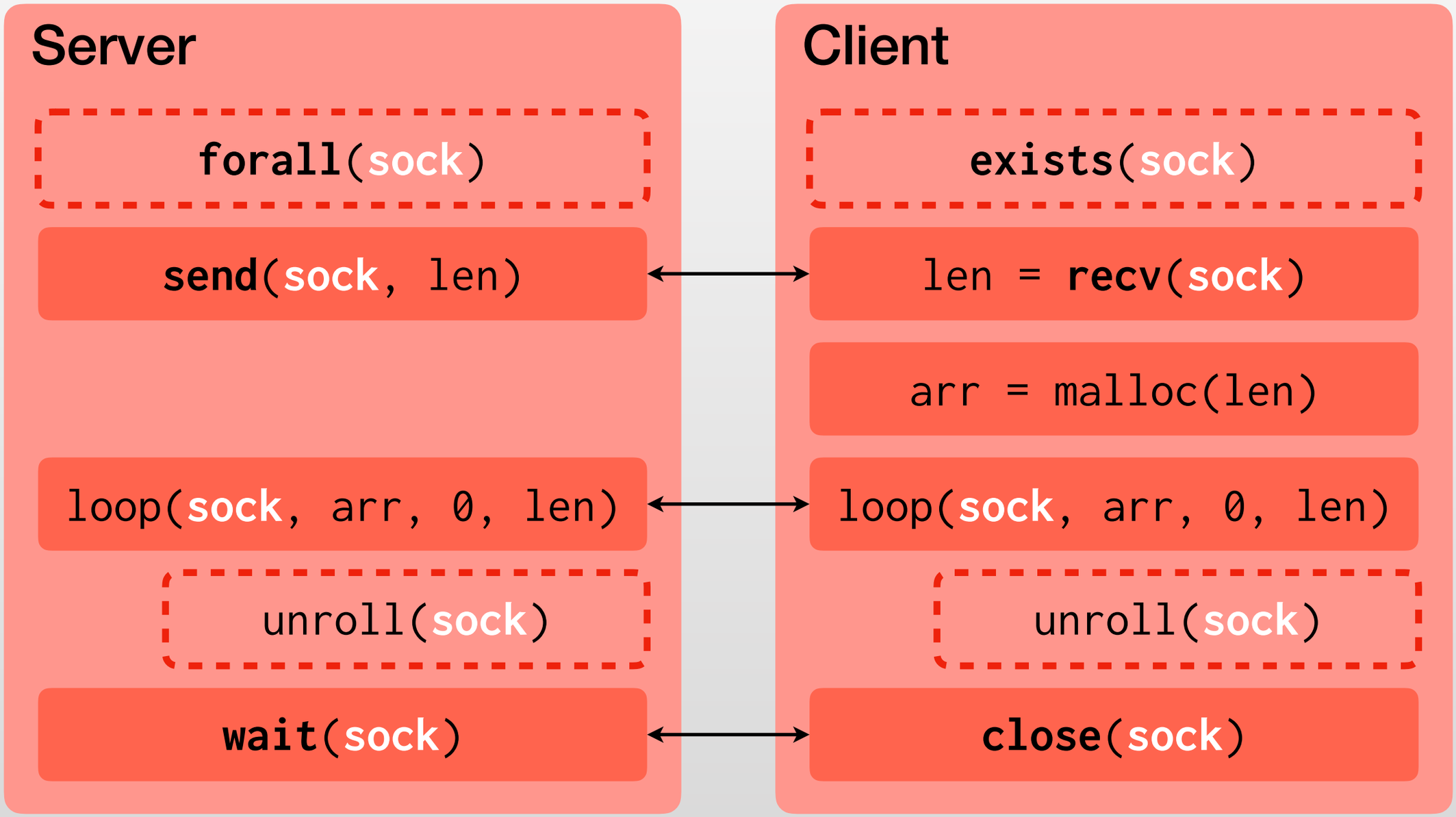
The socket is typed using simple session types.
Linear types guarantee correct order and no leak.
What else could go wrong?

Sending an Array, Dependent



The socket is typed using dependent session types.
Nothing could go wrong.

Sending an Array, Dependent



Show Me the Code

```
rpt(n,s) := fix(n,λf:int→stype.λn:int.ite(n>0,s::f(n-1),end(1)))
proto    := quan(S,λn:int.msg(S,int(n))::rpt(n,msg(S,int)))

server ∀n.int (ch:sock(S,proto), arr:int[n], len:int(n)) =
  let loop ∀m.int,m≤n (ch:sock(S,rpt(m,msg(S,int))), x:int(m)) =
    if x = 0 then
      unroll(ch); ite_false(ch); close(ch)
    else
      unroll(ch); ite_true(ch); send(ch,arr[len-x])
      loop(ch,x-1)
in
   forall(ch); send(ch,len); loop(ch,len)
end
```

Takeaway

untyped sockets

- runtime errors
- hard to debug
- deadlock
- resource leak

simple session types

- type errors
- correct use of sockets
- deadlock-free
- no leak

dependent session types

- more expressive
- quantifiers
- recursions
- polymorphic sessions

Even More

Formulatation
Lambda Calculus + Some

Logic Foundation
Multirole Logic, Cut-Free

Bi-directional Forwarding

Used in Classrooms

Higher-order Sessions

Implementations
Erlang, C, Javascript

Multiparty Sessions

Multi-directional Forwarding

Proof of Deadlock-Freeness

learn more at

multirolelogic.org

Thank you!

Hanwen Wu and Hongwei Xi

Dec 2017, IBM Programming Language Day