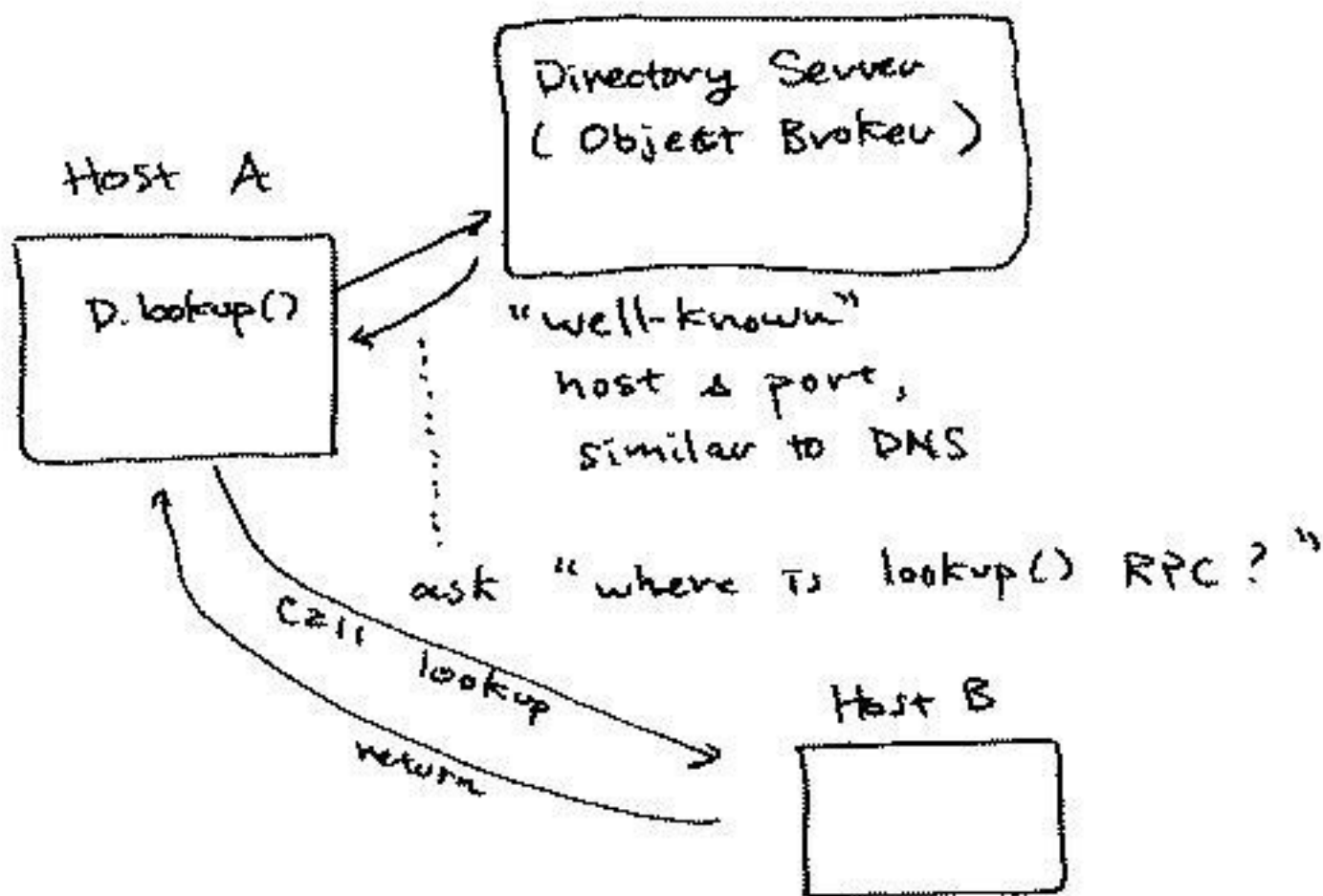


# Binding / Directory Service

Question: How does client know where server is?

remember, we want to hide networking from programmer using RPC

Answer: using directory



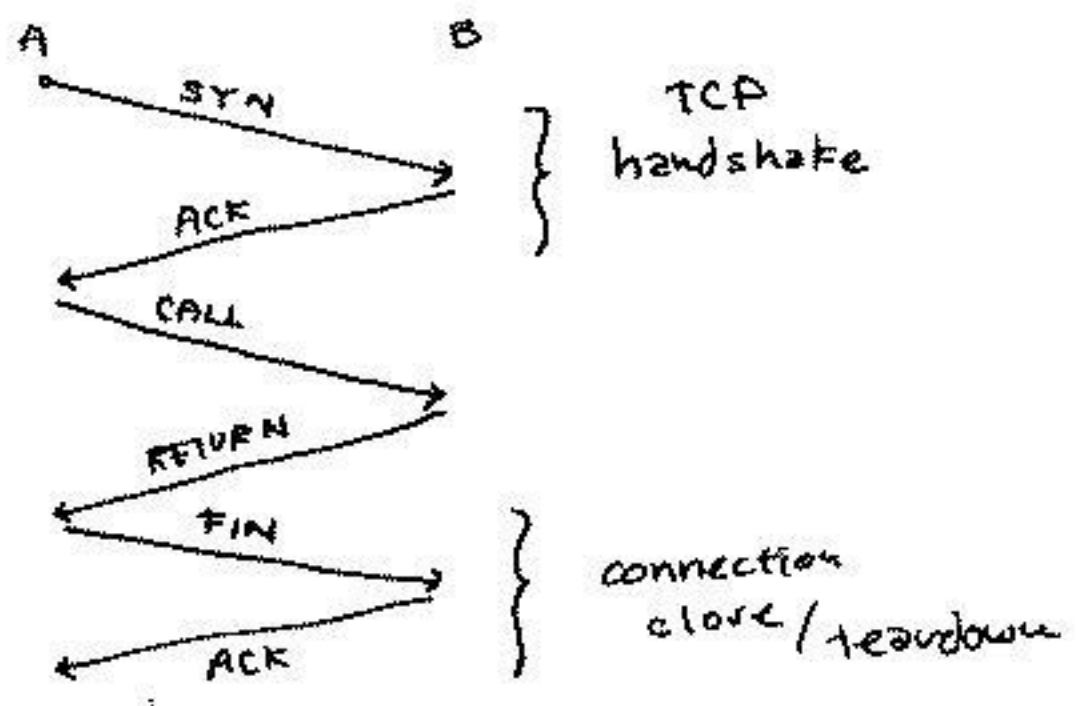
After discovery of lookup()'s location, future calls are "bound" to Host B.

Directory Server has several functions

- discover & bind
- register (new RPC methods)
- maybe some fault tolerance, security

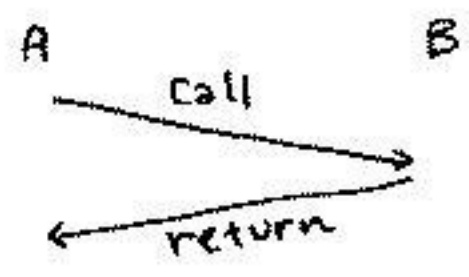
# RPC network implementation

▷ TCP



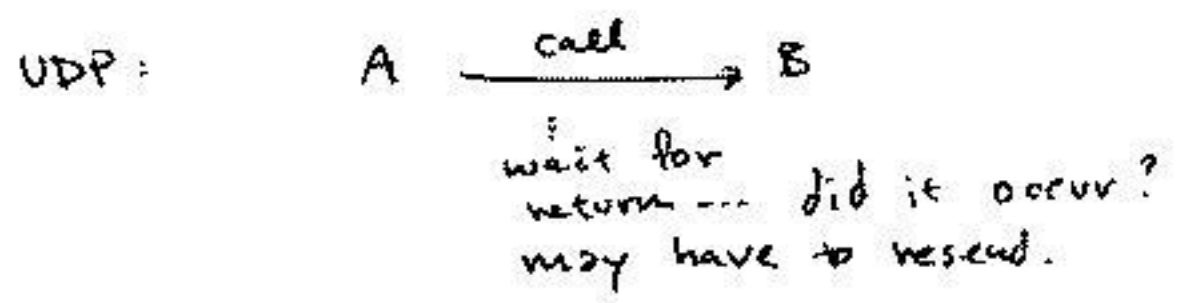
recall: TCP has reliability guarantees

▷ UDP



obviously lower overhead (but not stream-oriented)

What if there's an error?



## RPC Semantics

types of calls

Idempotent or non-idempotent.

function  $f$  is idempotent iff

$$\text{for any } x, f(x) = f(f(x))$$

Examples:

- add zero
- read content of a variable
- write 5 to variable  $y$

Not idempotent

- debit \$100 from bank account
- turn robot arm 3 degrees

## Call Semantic Requirements

(part of the description / specification / document)

- call exactly once
- call at most once
- call at least once

QUESTION: for which of these is

TCP or UDP appropriate?

## RPC and failures

Two kinds of failures (not counting bugs!)

(1) System / Hardware

messages lost, host crash/restart,  
connection lost

(2) Exceptions

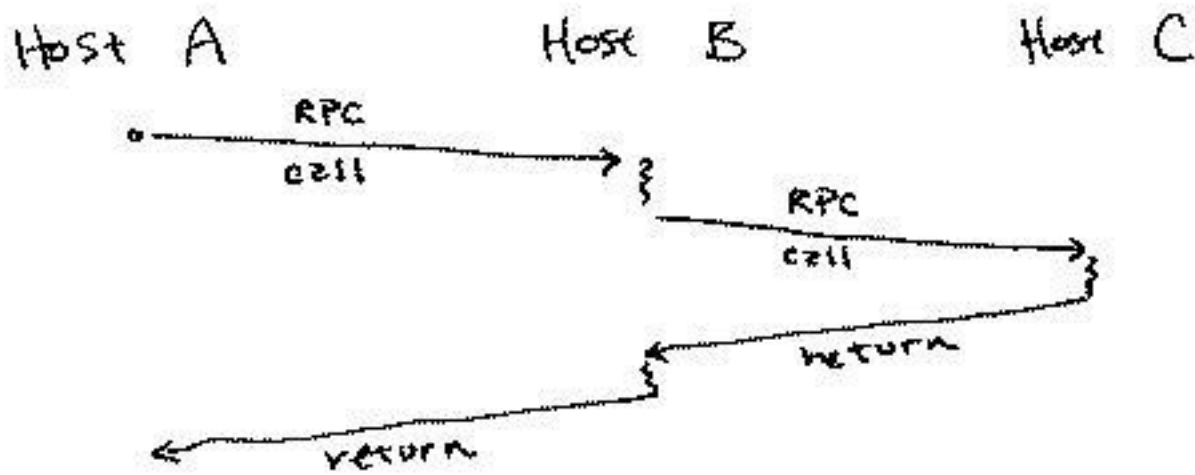
software-triggered

(1) Will be mostly handled by network protocol layers (plus extra steps to support RPC semantics of call)

(2) Exceptions are language-specific -- can always be treated as some kind of return result (but we will see this can be optimized sometimes)

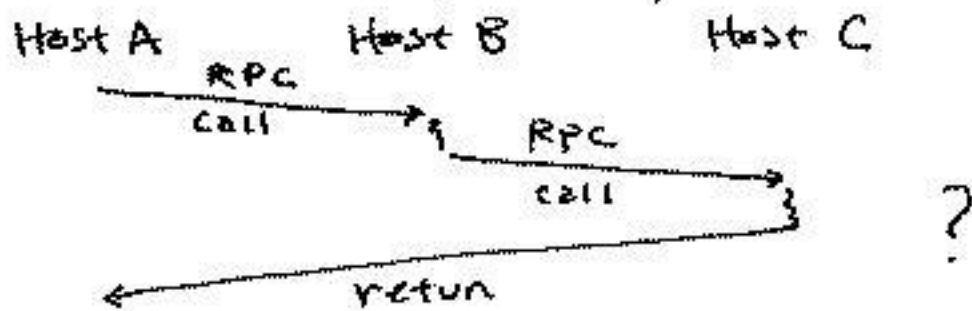
# NESTED RPC

It's quite possible:

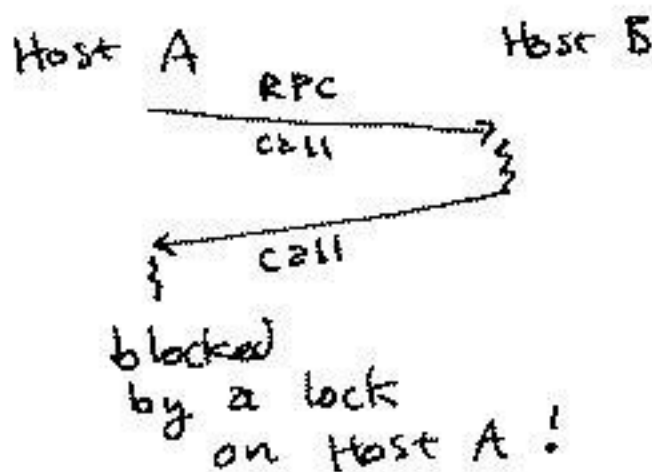


When nested RPC is possible, we need to think about

- security (not yet discussed in class)
- failures
- asynchronous RPC, example:



- locks and deadlocks



# RPC parameters

call by value? call by reference?

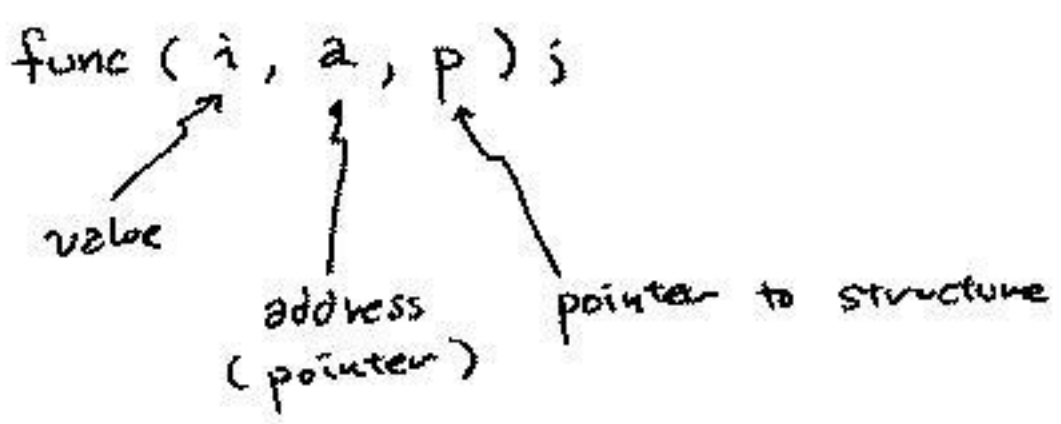
Let's look at RPC for C or C++

```

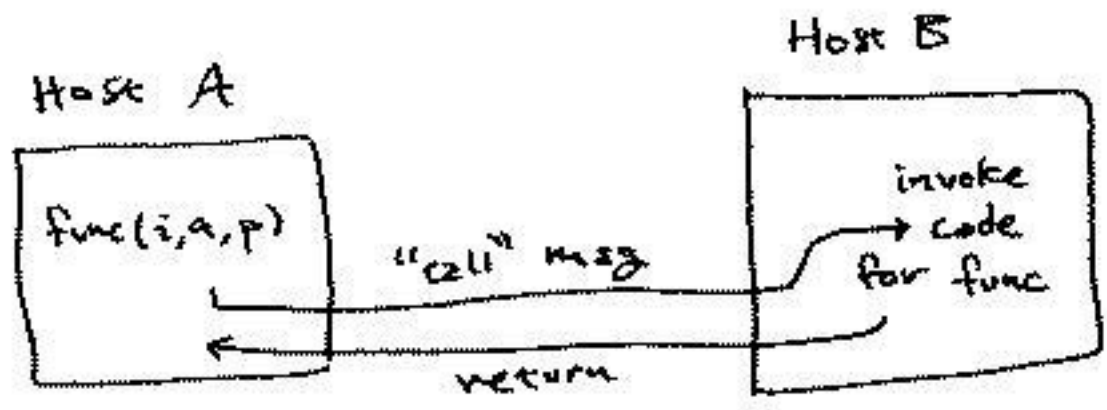
int i;
float a[5];
table * p; /* linked list */

```

DCE's  
RPC



## GOAL OF RPC



Question: what if func(i,a,p) changes some element of a?

pointers to memory space on Host A are meaningless on Host B.