

# RPC Model

Peter J. Denning

# A Procedure Interface

- Internet model is a relatively low-level interface with the protocol stack
  - User requesting a remote service has to encode a datagram, send it, wait for return datagram, decode it, and take the return value
- Similar to the protected service process model discussed earlier

# A Procedure Interface - 2

- Hide message passing protocol behind a simple interface that mimics procedure call
  - We did this before, for a local procedure call, with the PPC(p, args) interface
- Do the same for a remote procedure call, with the RPC(p, args) interface

# Operation

- $r = \text{RPC}(p, a)$ 
  - call service  $p$  with args  $a$
  - use name service to get IP address  $A$  for the machine hosting service  $p$
  - encode arguments  $a$  into a datagram to be sent to socket  $A/p$  (marshaling)
  - send the datagram and wait for response via local network service process
  - Decode, extract, and return  $r$
  - Resend datagram if no response in timeout period

# Operation - 2

- Extend RPC to handle local procedures as well by maintaining a list of local service processes
  - RPC checks that the desired service is on the local list
  - if so performs a PPC(p, args)
  - if not perform RPC(p, args) as above
- The code for this extended RPC is called a “stub” in the RPC literature
  - Compiler substitutes the stub for a procedure call in a program