DISTRIBUTED NETWORK PROGRAMMING

DILIP KUMAR SHRESTHA

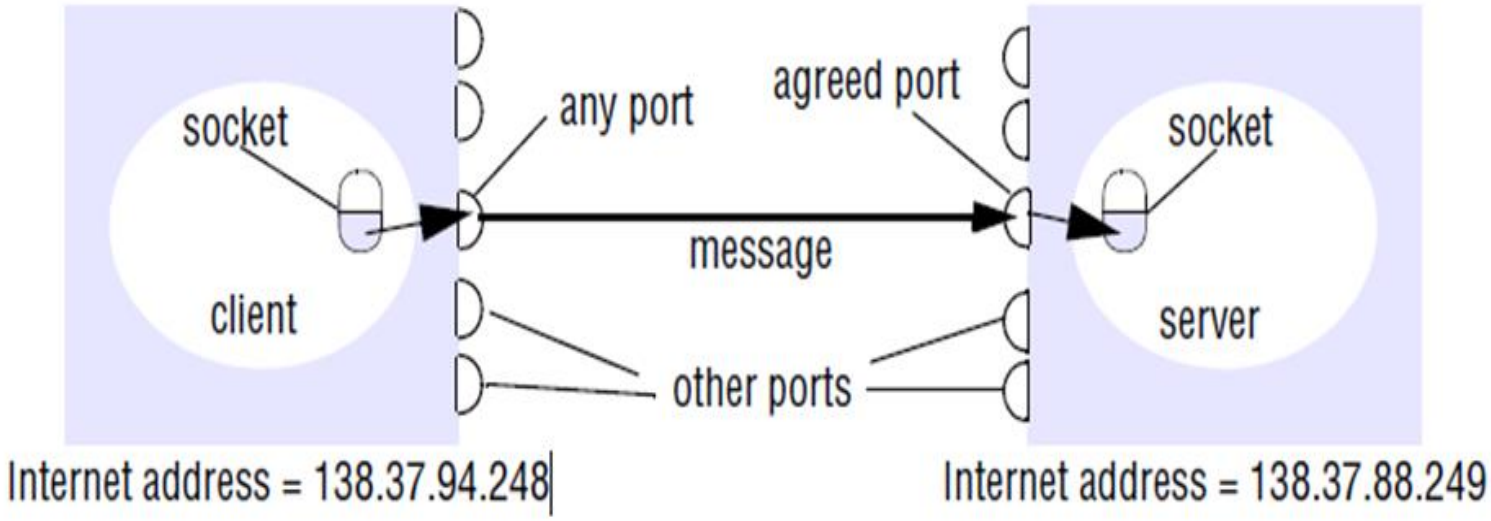
OVERVIEW

- Introduction
- The API for the Internet Protocol
- External data representation and marshalling
- Client server communication
- Group communication

THE API FOR THE INTERNET PROTOCOL

UDP

TCP



ISSUES

Datagram

- Packet size
- Blocking
- Timeouts
- Receive from any
- Failure model
  - Omission Failure
  - Ordering

Stream

- Message size
- Lost message
- Flow control
- Message duplication and ordering
- Message destinations
- Outstanding issues
  - Matching of data item(data format)
  - Blocking
  - Threads
- Failure Model
  - Ideal protocol

USAGES

Datagram

- DNS
- VOIP

Stream

- HTTP
- FTP
- Telnet
- SMTP

COBRA'S COMMON DATA REPRESENTATION(CDR)

Primitive Types

Type	Size(bit)
Short	16
Long	32
Unsigned short	16
Unsigned long	32
Float	32
Double	64
Char	1
Boolean	1
Octet	8

Composite/Constructed Types

Type	Representation
Sequence	Length(unsigned long) followed by elements in order
String	Length(unsigned long) followed by characters in order
Array	Array elements in order(fixed length)
Struct	In the order of declaration of the components
Enumerated	Unsigned long(the values are specified by the order of declaration)
Union	Type tag followed by the selected member

EXAMPLE

struct Person{
string name;
string place;
unsigned long year;
};

Let us consider Person struct with value:

{'Smith', 'London', 1984}.

index in sequence of bytes	←4 bytes→	*notes on representation*
0–3	5	*length of string*
4–7	"Smit"	*'Smith'*
8–11	"h___"	
12–15	6	*length of string*
16–19	"Lond"	*'London'*
20–23	"on__"	
24–27	1984	*unsigned long*

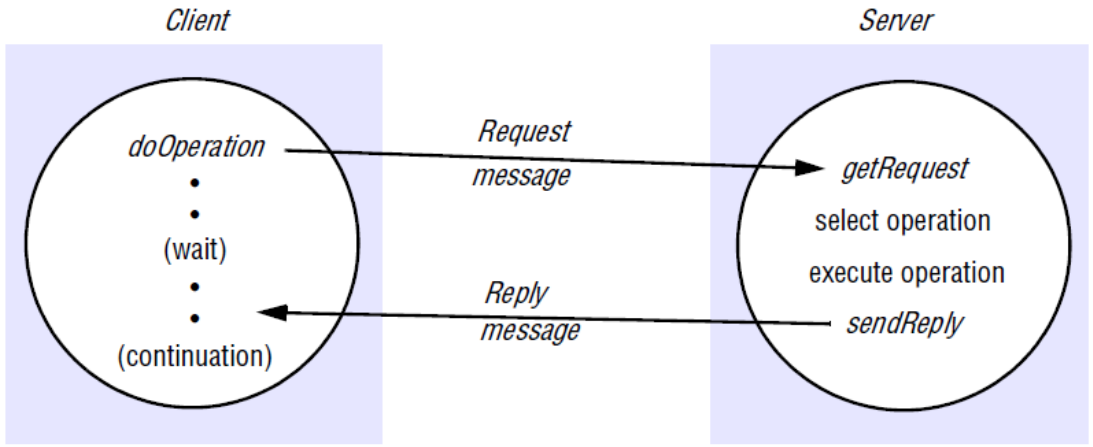
REMOTE OBJECT REFERENCE

A *remote object reference* is an identifier for a remote object that is valid throughout a distributed system. A remote object reference is passed in the invocation message to specify which object is to be invoked.

Internet Address	Port number	Time	Object number	Interface of remote object
32 bits	32 bits	32 bits	32 bits	

CLIENT SERVER COMMUNICATION

Request-Reply protocol primitives



Request-reply message structure

messageType	*int (0=Request, 1= Reply)*
requestId	*int*
remoteReference	*RemoteRef*
operationId	*int or Operation*
arguments	*// array of bytes*

COMMUNICATION BETWEEN DISTRIBUTED OBJECTS

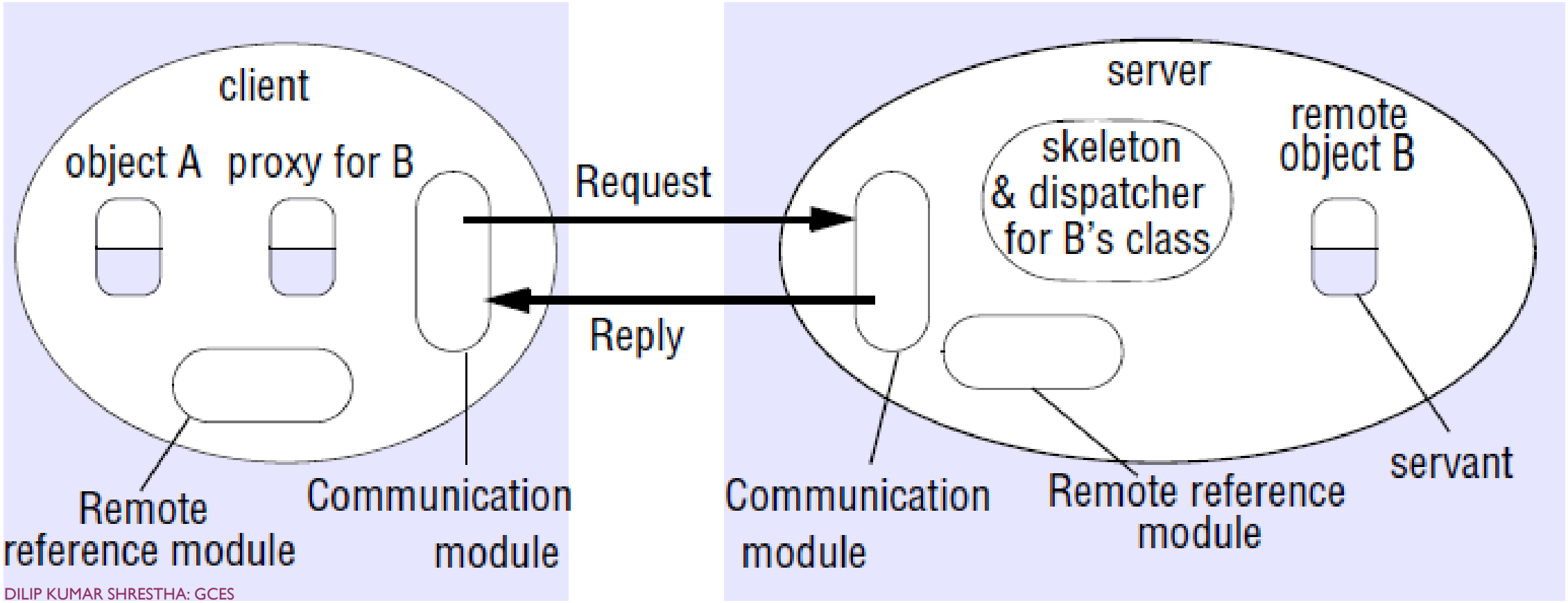
Object Model	Remote Object Model
Object reference	Remote Object reference
Interface	Remote Interface
Actions	Actions
Exception	Exception
Garbage collection	Garbage collection

REMOTE METHOD INVOCATION

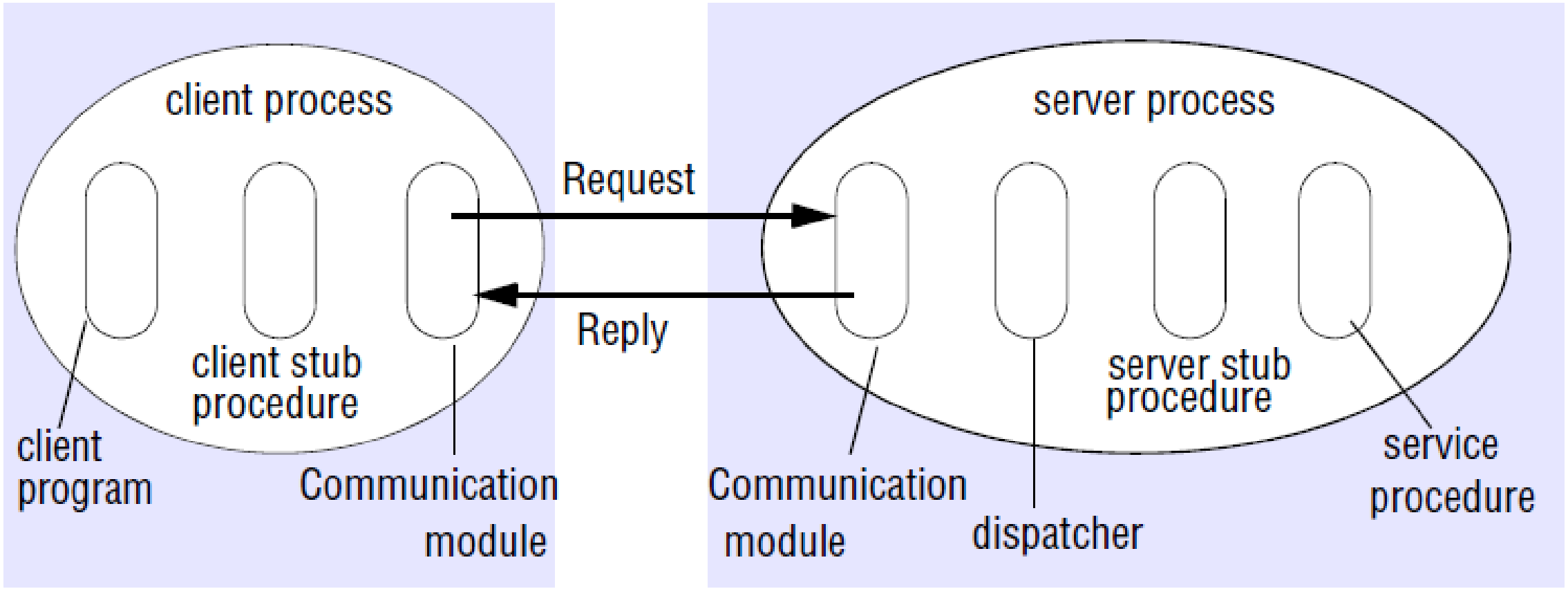
Design Issues

- RMI Invocation Semantics
  - Maybe
  - At-least-once
  - At-most-once
- Transparency
  - Syntax and semantics
  - Latency
  - Exception

RMI IMPLEMENTATION



RPC IMPLEMENTATIONS



CORBA IMPLEMENTATION