## Resumo

Remote Procedure Call (RPC) is a [protocol](https://searchnetworking.techtarget.com/definition/protocol) that one program can use to request a service from a program located in another computer on a [network](https://searchnetworking.techtarget.com/definition/network) without having to understand the network's details. RPC is used to call other processes on the remote systems like a local system. A procedure call is also sometimes known as a *function call* or a *subroutine call*.

RPC uses the [client-server](https://searchnetworking.techtarget.com/definition/client-server) model. The requesting program is a client, and the service-providing program is the [server](https://whatis.techtarget.com/definition/server). Like a regular or local procedure call, an RPC is a [synchronous](https://whatis.techtarget.com/definition/synchronous) operation requiring the requesting program to be suspended until the results of the remote procedure are returned. However, the use of lightweight processes or [threads](https://whatis.techtarget.com/definition/thread) that share the same address space enables multiple RPCs to be performed concurrently.

The interface definition language (IDL) -- the specification language used to describe a software component's application programming interface (API) -- is commonly used in Remote Procedure Call software. In this case, IDL provides a bridge between the machines at either end of the link that may be using different operating systems (OSes) and computer languages.

### **RPC message procedure**

When program statements that use the RPC framework are compiled into an executable program, a [stub](https://whatis.techtarget.com/definition/stub) is included in the compiled code that acts as the representative of the remote procedure code. When the program is run and the procedure call is issued, the stub receives the request and forwards it to a client [runtime](https://searchsoftwarequality.techtarget.com/definition/runtime) program in the local computer. The first time the client stub is invoked, it contacts a name server to determine the transport address where the server resides.

The client runtime program has the knowledge of how to address the remote computer and server application and sends the message across the network that requests the remote procedure. Similarly, the server includes a runtime program and stub that interface with the remote procedure itself. Response-request protocols are returned the same way.

### **RPC models and alternative methods for client-server communication**

There are several RPC models and [distributed computing](https://whatis.techtarget.com/definition/distributed-computing) implementations. A popular model and implementation is the Open Software Foundation's ([OSF](https://whatis.techtarget.com/definition/Open-Software-Foundation-OSF)) Distributed Computing Environment ([DCE](https://searchnetworking.techtarget.com/definition/DCE)). The Institute of Electrical and Electronics Engineers ([IEEE](https://whatis.techtarget.com/definition/IEEE-Institute-of-Electrical-and-Electronics-Engineers)) defines RPC in its ISO Remote Procedure Call Specification, ISO/IEC CD 11578 N6561, ISO/IEC, November 1991.

RPC spans the [transport layer](https://searchnetworking.techtarget.com/definition/Transport-layer) and the [application layer](https://searchnetworking.techtarget.com/definition/Application-layer) in the Open Systems Interconnection ([OSI](https://searchnetworking.techtarget.com/definition/OSI)) model of network communication. RPC makes it easier to develop an application that includes multiple programs distributed in a network.

Alternative methods for client-server communication include [message queueing](https://searchapparchitecture.techtarget.com/definition/message-queueing) and IBM's Advanced Program-to-Program Communication ([APPC](https://searchnetworking.techtarget.com/definition/APPC)).

### **How does RPC work?**

When a remote procedure call is invoked, the calling environment is suspended, the procedure parameters are transferred across the network to the environment where the procedure is to execute, and the procedure is then executed in that environment.

When the procedure finishes, the results are transferred back to the calling environment, where execution resumes as if returning from a regular procedure call.

During an RPC, the following steps take place:

1. The client calls the client stub. The call is a local procedure call with parameters pushed onto the stack in the normal way.
2. The client stub packs the procedure parameters into a message and makes a system call to send the message. The packing of the procedure parameters is called [*marshalling*](https://whatis.techtarget.com/definition/marshalling).
3. The client's local OS sends the message from the client machine to the remote server machine.
4. The server OS passes the incoming packets to the server stub.
5. The server stub unpacks the parameters -- called *unmarshalling* -- from the message.
6. When the server procedure is finished, it returns to the server stub, which marshals the return values into a message. The server stub then hands the message to the transport layer.
7. The transport layer sends the resulting message back to the client transport layer, which hands the message back to the client stub.
8. The client stub unmarshalls the return parameters, and execution returns to the caller.

### **Types of RPC**

There are five types of RPC:

1. The normal method of operation where the client makes a call and doesn't continue until the server returns the reply.
2. The client makes a call and continues with its own processing. The server doesn't reply.
3. A facility for sending several client nonblocking calls in one batch.
4. RPC clients have a broadcast facility, i.e., they can send messages to many servers and then receive all the resulting replies.
5. The client makes a nonblocking client/server call; the server signals the call is completed by calling a procedure associated with the client.

### **Advantages of Remote Procedure Call**

The advantages of Remote Procedure Call include the following:

* helps clients communicate with servers via the traditional use of procedure calls in high-level languages;
* can be used in a distributed environment, as well as the local environment;
* supports process-oriented and thread-oriented models;
* hides the internal message-passing mechanism from the user;
* requires only minimal effort to rewrite and redevelop the code;
* provides [abstraction](https://whatis.techtarget.com/definition/abstraction), i.e., the message-passing nature of network communication is hidden from the user; and
* omits many of the protocol layers to improve performance.

### **Disadvantages of RPC**

Some of the disadvantages of RPC include the following:

* The client and server use different execution environments for their respective routines, and the use of resources -- e.g., files -- is also more complex. Consequently, RPC systems aren't really suited for transferring large amounts of data.
* RPC is highly vulnerable to failure because it involves a communication system, another machine and another process.
* There is no uniform standard for RPC; it can be implemented in a variety of ways.
* RPC is only interaction-based, and as such, it doesn't offer any flexibility when it comes to hardware architecture.