EDOC018

EUROGAM PROJECT

NSF DATA ACQUISITION SYSTEM

Remote Procedure Call

Edition 2.0
July 1990

NSF Software Systems Group
UK Science and Engineering Research Council
Daresbury Laboratory

Introduction

This document describes the protocol specification for a Remote Procedure Call mechanism forming part of the NSF Data Acquisition system.

The protocol is a compatible subset of the RPC protocol defined by Sun Microsystems.

This document describes Version 2 of the RPC protocol.

This protocol will be carried by a datagram type communications protocol such as described in the document *Ethernet Communications.*

The protocol dialogue is between an INITIATOR and a TARGET. Only the INITIATOR may send protocol requests, the TARGET will reply with a protocol reject or accept.

RPC messages passed by a datagram transport are treated as unreliable. Thus if the initiator retransmits call messages after suitable time-outs the only thing that can be inferred from no reply message is that the remote procedure was executed zero or more times and from a reply message that it was executed one or more times.

The initiator application may have a number of requests outstanding at any time which can be identified by use of the *transaction id.*

All error detection and recovery is the responsibility of the application users of the protocol with the exception of those errors reported by the reject message.

Program interface

The Sun Microsystems RPC library may be used for writting application client and server programs if the constraints and limitations placed by use of the library routines are acceptable. Use of the library routines is however not essential as long as the protocol is followed.

For full details see the Sun manual *Remote Procedure Call - Programming Guide.* For additional information see also the Eurogam document *RPC Applications Programs.*

Protocol Specification

The protocol diagrams are in terms of data items 32 bits wide.

REQUEST

General form of a RPC request.

transaction id
message type : call = 0
RPC version = 2
program
version
procedure
authorisation
procedure specific data

Transaction id is a 4 byte field available for use of the client to identify this REQUEST. This field is returned with the subsequent RESPONSE. It is available for clients to match reply messages with call messages. The server cannot treat this ID as any type of sequence number.
Program, *Version* and *Procedure* specify the remote program, its version number, and the procedure within the remote program to be called.
Authorisation is a field which contains the client's authorisation to access the server. Currently only the null authorisation is used and hence this field is 16 bytes long and should be set to nulls.
Procedure specific data contains application program specific request data which defines the parameters to be passed to the called procedure.

RESPONSE

The response to a RPC request can take one of two forms: the message was either accepted or rejected.

Reject

The RPC REQUEST may be rejected by the server.

transaction id
message type : reply $= 1$
reply status : reject $= 1$
reason

The request can be rejected by the server because of two reasons: either the server is not running a compatible version of the RPC protocol or the server refuses to accept the client's authorisation .
Reason is a 4 byte field which indicates the reason for the reject.

reason=	0	RPC version number invalid (ie not equal 2)
	1	Authorisation failure

Since only RPC version 2 is acceptable and only the null authorisation is used these errors are not expected.

Accept - Error

The RPC REQUEST may be accepted by the server but subsequentially an error detected.

transaction id
message type : reply $= 1$
reply status : accept $= 0$
authorisation verifier
reason

Authorisation verifier is a field returned by the server in response to the client's authorisation information. Since only the null authorisation is currently used this field is 8 bytes long and should be set to nulls.
Reason is a 4 byte field which is non zero and indicates the cause of the error.
Note *reason* $= 0$ indicates successful execution of the procedure.

reason=	1	remote program invalid or unavailable
	2	remote program version invalid or not supported
	3	remote procedure invalid or not supported
	4	precedure cannot decode parameters

Accept - OK

The RPC REQUEST may be accepted by the server and executed successfully.

transaction id
message type : reply = 1
reply status : accept = 0
authorisation verifier
accept status : success = 0
procedure specific data

Authorisation verifier is a field returned by the server in response to the client's authorisation information. Since only the null authorisation is currently used this field is 8 bytes long and should be set to nulls.
Procedure specific data contains application program specific response data which defines the results to be returned to the calling program.