RPC

Enrico Deiana, Emanuele Del Sozzo

1 rpc.h

The global variables declared here are:

- extern PacketEncoder *rpcInterfaceShadowDaemonPacketEncoder
- extern PacketEncoder *rpcInterfaceTerminalPacketEncoder
- extern PacketEncoder *rpcInterfaceKernelPacketEncoder
- extern EtnNullEncoder *rpcInterfaceNullEncoder
- extern EtnBufferDecoder *rpcInterfaceBufferDecoder
- extern EtnRpcHost *rpcInterfaceShadowDaemonHost
- extern EtnRpcHost *rpcInterfaceTerminalHost
- extern EtnRpcHost *rpcInterfaceKernelHost
- extern EtnRpcHost *rpcInterfaceNullHost

Using extern keyword, C variables are declared but not defined. So, they must be defined in another header or c file before using them. In particular, these variables are defined in *packetEncoder.c* Moreover, the extern extends the visibility to the whole program. These global variables are used as hosts, encoders and decoders of Ethos primary components.

The functions exported by the header file are:

- *void rpcInit(void)*
  This function calls some other functions that initialize rpc components. The definition is inside *rpc.c*.
- *rpcCall(fn, host, connection, eventId, ...)*
  This macro is used to:
  1. calculate length of encoded packet with a dummy run
  2. reset the packet encoder with this length
  3. perform the actual RPC call

2 rpc.c

The global variable declared and defined here is:

- DebugFlagDesc debugFlagDesc[]
  This is simply an array of debug flags. It is in *rpc.c* because it is used from both *Ethos* and *Dom0* even though it is not RPC-specific.

In this file, there are some debug purpose functions. They are used to send a ping between both Shadow-daemon and Terminal. Here they are:

- *void rpcShadowDaemonPing (EtnRpcHost *h, uint64_t eventId)*

- *void rpcShadowDaemonPingReply (EtnRpcHost *h, uint64_t eventId, Status status)*
  Parameters *h* and *eventId* are useless in this function since they are not used.

- *void rpcTerminalPing (EtnRpcHost *h, uint64_t eventId)*

- *void rpcTerminalPingReply (EtnRpcHost *h, uint64_t eventId, Status status)*
  Parameters *h* and *eventId* are useless in this function since they are not used.

The other functions defined here are:

- *void rpcInitInterfaces(void)*
  This function initialize the interfaces. All the variables that are initialized are global variables declared in *rpc.h* and defined in *packetEncoder.c*. For each Ethos primary component a packet encoder is defined. Then, a null encoder and a buffer decoder are also defined. These five interfaces, in addiction to the servers of each primary component, are used to define the interface host of each primary component. This function contains very bad casts since they are memory allocation dependent.

- *void rpcInit(void)*
  This function calls some other functions that initialize rpc components. The declaration is inside *rpc.h*.

3 packetEncoder.h

The data structure defined here is *packetEncoder*. It contains:

- EtnBufferEncoder encoder
- Packet *packet
- uint32_t totalLength
- bool packetSent
- Connection *connection

The functions declared here are:

- *PacketEncoder *packetEncoderNew(void)*
  This function creates a new, blank packetEncoder. Must be reset before use. The definition is inside *packetEncoder.c*.

- *void packetEncoderReset(PacketEncoder *e, Connection *c, uint32_t totalLength)*
  This function resets the packetEncoder. The definition is inside *packetEncoder.c*.

- *int packetEncoderWrite(EtnEncoder *_e, uint8_t *data, uint32_t length)*
  This function writes data to a packet. Inside *packetEncoder.c* there is the definition of *_packetEncoderWrite* which is *static*. By definition of *static* keyword, a function declared *static* cannot be seen outside the file it was defined in. So, if they are the same function, it is not clear why it is also defined in header file.

- *void packetEncoderFlush(EtnEncoder *e)*
  This function flushes packet encoder, sending the packet as the final fragment. Inside *packetEncoder.c* there is the definition of *_packetEncoderFlush* which is static. By definition of *static* keyword, a function declared *static* cannot be seen outside the file it was defined in. So, if they are the same function, it is not clear why it is also defined in header file.

4 packetEncoder.c

The global variables declared in *rpc.h* are defined here, although *rpcInterfaceNullEncoder* variable here has type *PacketEncoder*, while in file *rpc.h* it has type *EtnNullEncoder*.

The functions defined here are:

- *static void _packetEncoderFlush (EtnEncoder *_e)*
  This function flushes packet encoder, sending the packet as the final fragment.
  This function contains a very bad cast since it is memory allocation dependent.
- *static void _packetEncoderCreateMax (PacketEncoder *e, bool firstPacket)*
  This function creates a packet of the maximum size allowed by the corresponding tunnel. It will increment sequence counter; if a packet is created here it must be sent.
- *static int _packetEncoderWrite (EtnEncoder *_e, uint8_t *data, uint32_t length)*
  This function writes data to a packet, sending fragments as maximum size reached. The final (non-full) fragment is not sent because there may be remaining data to write to it. This fragment is sent when the flush function is called. Note that if the write does not fill the first fragment, then nothing will be sent until flush is called.
  This function contains a very bad cast since it is memory allocation dependent.
- *void packetEncoderReset (PacketEncoder *e, Connection *c, uint32_t totalLength)*
  This function resets the packetEncoder. The declaration is inside *packetEncoder.h*.
  This function contains a very bad cast since it is memory allocation dependent.
- *PacketEncoder *packetEncoderNew()*
  This function creates a new, blank packetEncoder. Must be reset before use. The declaration is inside *packetEncoder.h*.

5 connection.h

6 connection.c

7 erpc.c

8 etn.h