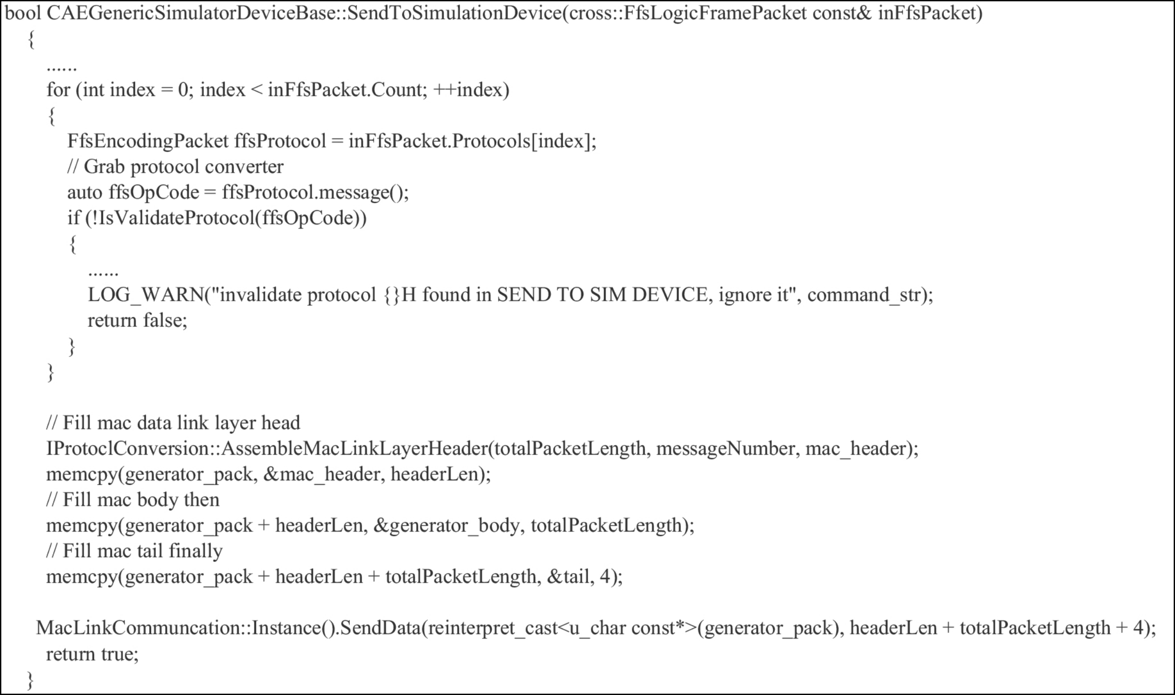

```
bool CAEGenericSimulatorDeviceBase::SendToSimulationDevice(cross::FfsLogicFramePacket const& inFfsPacket)
  {
    ......
    for (int index = 0; index < inFfsPacket.Count; ++index)
    {
      FfsEncodingPacket ffsProtocol = inFfsPacket.Protocols[index];
      // Grab protocol converter
      auto ffsOpCode = ffsProtocol.message();
      if (!IsValidateProtocol(ffsOpCode))
      {
        ......
        LOG_WARN("invalidate protocol {}H found in SEND TO SIM DEVICE, ignore it", command_str);
        return false;
      }
    }

    // Fill mac data link layer head
    IProtoclConversion::AssembleMacLinkLayerHeader(totalPacketLength, messageNumber, mac_header);
    memcpy(generator_pack, &mac_header, headerLen);
    // Fill mac body then
    memcpy(generator_pack + headerLen, &generator_body, totalPacketLength);
    // Fill mac tail finally
    memcpy(generator_pack + headerLen + totalPacketLength, &tail, 4);

   MacLinkCommuncation::Instance().SendData(reinterpret_cast<u_char const*>(generator_pack), headerLen + totalPacketLength + 4);
    return true;
  }
```