Invited Talk:

Mitigating the Effects of Internet Timing Faults Across Embedded Network Gateways

Philip Koopman, Justin Ray

Carnegie Mellon University, ECE Department, 5000 Forbes Ave.
Pittsburgh, PA 15213, USA
koopman@cmu.edu, justinr2@cmu.edu

Extended Abstract

Traditional embedded systems such as automobiles and industrial controls are increasingly being connected to enterprise computing facilities and the Internet. The usual approach to making such a connection is to install a *gateway* node which translates from Internet protocols to embedded field bus network protocols. Such connections raise obvious security concerns, because the gateway must guard against attacks on the embedded devices it serves. For our purposes, we'll assume that typical enterprise and Internet vulnerabilities, such as buffer overflows, have already been taken care of. (Securing devices against traditional attacks is no small matter, but we are interested in uniquely embedded issues.)

Beyond normal gateway functions, an Internet to embedded gateway must also prevent timing faults and timing attacks from crossing over the gateway to affect the operation of attached embedded systems. An example of timing fault propagation would be severe clumping of messages on the Internet side so that many messages arrive at the gateway all at once, disrupting embedded system operation. While a queue can reduce the loss of incoming data and mitigate network overload, it cannot necessarily protect against timing-related faults on the embedded side of the gateway.

We report simulation results for several mechanisms to mitigate the effects of Internet message timing variations (whether due to faults or malicious attacks) on the performance of networked embedded systems using real-time data. Problems are caused primarily by excessive data delivery delay rather than messages being dropped from arriving clumps. This means that putting a queue in the gateway to manage arriving data clumps is typically worse than using no mitigation mechanism at all. Using a predictive filter seems intuitively better than using a queue, but finding a good generalized predictive filter is also quite difficult.

We believe that managing data streams from the Internet to embedded systems will require careful attention to the nature and time constants of data flowing through the gateway. Moreover, it seems likely that each distinct data stream will need a different set of data management mechanisms and policies at the gateway. In this case, one size *does not* fit all, making the design of a robust gateway a difficult problem that will require careful modeling of data value behavior for every gateway built.

Keywords: Embedded network, gateway, embedded security, timing fault, simulation, predictive filter