Loss And Error Encoding

Perry Kundert

2014-06-25 00:00:00

Real-time communications over quite reliable channels (eg. loss/error rates $< 50\%$) is quite well researched, and is easily accessible.

However, the maintenance of reliable, deterministic two-way communications over multiple degenerate channels (eg. loss/error rates $>50\%$, uni-directional communications, arbitrarily long latencies, hostile adversaries able to record and replay communications) is much more difficult, and commercial solutions are not as readily available. (PDF / Text)

Contents

1 Solution **1**
1.1 Loss and Error Coding 1
1.2 Determinism 2
1.3 Embedded Systems 2

1 Solution

I have implemented and deployed a solution on behalf of a very large client, which includes a novel communication protocol implemented between dozens of SCADA (Supervisory Control and Data Acquisition) control workstations, and hundreds of remote RTU nodes deployed continent wide.

1.1 Loss and Error Coding

Using Reed-Solomon (and other) loss and error coding schemes, we are able to dynamically add and suspend communications routes in an automated fashion to mitigate communications failures. Because the design of the protocol ensures that adding a new route can only reduce the communications channel risk, it can be done in a fully automated fashion, reducing the cognitive load on the operator and allowing them to instead focus on maintaining their mental model of the vast and complex hydrocarbon transport system they are controlling.

1.2 Determinism

Maintaining determinism was a priority. Typically, bidirectional channels over nondeterministic media like Ethernet (eg. TCP/IP) provide no determinism guarantees. In other words, in degenerate cases, control signals (eg. Close Valve, Start Pump) can be delayed for arbitrarily long periods of time, and delivered to the equipment with catastrophic effects. This is (of course) unacceptable, and means to retain deterministic deadlines for control and telemetry were implemented. This is an interesting challenge in a distributed system free of shared clocks...

1.3 Embedded Systems

Linux was selected as the embedded OS of choice, and has been in place for the better part of a decade controlling one of the largest and most complex continent-wide hydraulic systems on the planet with great success. This may not seem like a significant thing today, but at the time this choice was by no means as clear as it is today.