Hot Standby Bridges

Edward Hague, Chief Technology Officer, FieldServer Technologies

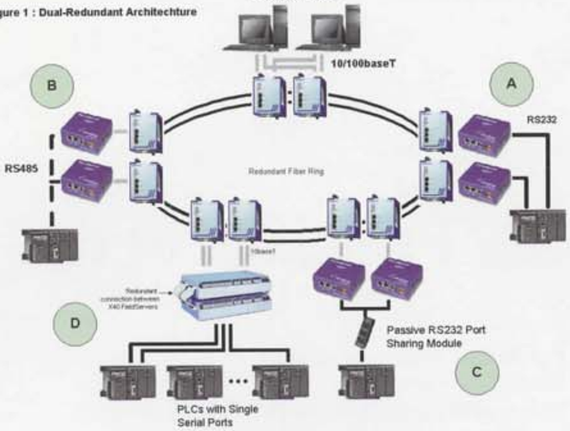
Mission critical applications require redundancy to operate reliably. With various serial devices interfacing to Ethernet backbones it is a challenge to develop a system robust enough to keep data flowing even with multiple connection failures.

There are a number of different physical layouts for a dual redundant Ethernet network (see Figure 1). At the top of this diagram is the host application (SCADA, HMI, PLC). The two ovals represent two redundant Ethernet segments, though whether this is achieved using ordinary or specialized network hardware is not relevant.

Subsystem A has two hubs, one on each Ethernet segment, connected to two remote serial ports (RSPs) which in turn connect to two serial ports on a single PLC. The PLC could be a single-chassis hot-standby PLC or two separate PLCs which operate in a hot-standby arrangement. This is straightforward to implement if the PLCs have two serial ports, are truly redundant **and** if the host application switch from one IP address to another transparently and automatically. Resolving these 'ifs' is the real subject of this article.

Many PLCs only have single serial ports such as in the **C** configuration of Figure 1. It is assumed here that if the PLC fails, the operation of the whole system is not compromised – possibly due to another system taking over the control responsibility, or perhaps the shutdown would only be localized to one small area in the plant. Regardless, if one of the networks failed, the data is still required to make it back to the host application.

Figure 1 : Dual-Redundant Architechture



The RS232 ports of the 2 bridges are routed via a RS232 splitter device that allows them to be connected to the single RS232 port of the PLC. Splitters are robust which drastically reduce the chances of failure. The problem is that the two bridges have to take turns at polling the PLC, since only one can poll at a time. There are PLCs which operate in a hot-standby mode with two serial ports, but the standby serial port is inactive until the standby PLC takes over from the original master. This arrangement is best dealt with using a virtual bridge arrangement mentioned later.

Subsystem B in figure 1 is implemented using RS485 that connects the two bridges to a single PLC. Although the physical connections are different, the logical arrangement and the operational requirements are exactly the same as in the system depicted in C.

The ideal solution is indicated in Subsystem D which shows FS-B40 serial bridges that do the RS232 splitting internally, and the single serial port on the PLC is shared between the two bridges via ribbon cables at the side. The FS-B40 can support up to 10 simultaneous connections, and up to hundreds of devices.

Token Passing

Bridge activity has to be orchestrated to avoid serial collisions. This can be achieved with a matched-pair of hot-standby bridges. Both operate identically, but they take turns at polling the PLC using a token-passing method. The token is transferred over Ethernet. The bridge with the token is allowed to poll the PLC on its serial port. When the response is received, the token is sent to the other bridge which then takes its turn. Timeouts, lost or duplicate tokens, or failed bridges are handled in an elegant fashion. If the two Ethernet segments are separate, then each bridge needs to be connected with each segment, requiring 2 Ethernet ports on each bridge.

SCADA/HMI unable to switch IP Address

Most of the arrangements discussed require that the host application is aware that data will stop being served from one of the bridges if the PLC connected to that bridge fails, or if the bridge itself fails. Unfortunately this is not always the case, but the problem can be solved in two ways, either using virtual bridges or dynamic IP addressing. Virtual bridges are discussed below. Dynamic IP addressing is relevant if the whole bridge fails. If bridge A fails, then bridge B takes over at the

Hot Standby Bridges

same IP address as bridge A, though this requires a virtual socket to be implemented in the bridges.

Virtual Bridges

Figure 2 shows a situation where two cables or serial ports have failed. The failed connection between Bridge 1 and PLC 1 on its own would not have caused problems, because the host application could still get the data from bridge 2. However, failure of the connection between the host Application and bridge 2 results in lost connectivity.

It is possible to use software-only protection against this failure. Understand that although a single point of failure tolerance is possibly adequate for some projects, it seems that adding a software function to protect against further failure modes is very desirable.

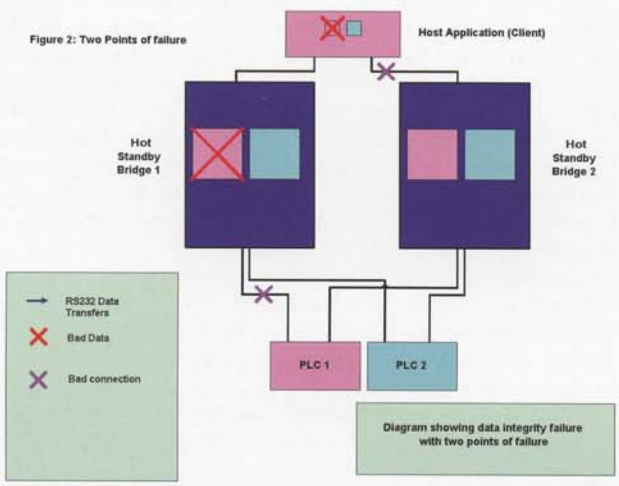


Figure 2: Two Points of failure

As can be seen in Figure 3, the physical connections are exactly the same as Figure 2. However the system is tolerant to 3 points of failure! Note that although this diagram shows the host application connected via RS232, the same principal applies to host applications that are connected via Ethernet.

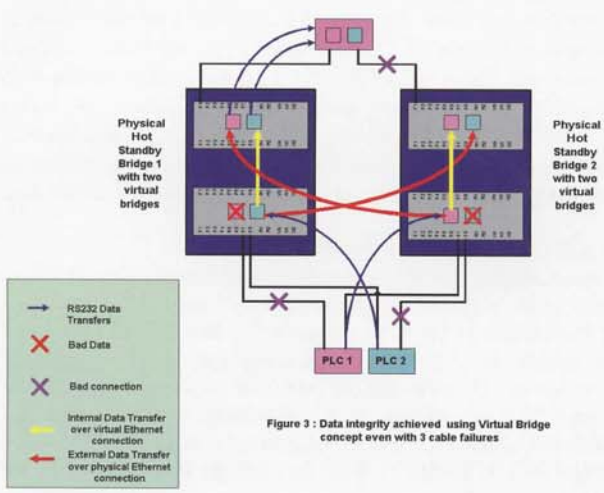
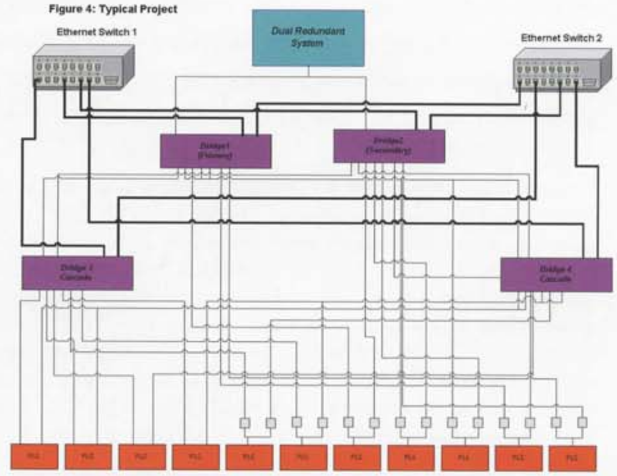


Figure 3 : Data integrity achieved using Virtual Bridge concept even with 3 cable failures

The top 'Virtual Bridge' will either fetch data from its own bottom Virtual Bridge or from the matching pair's bottom Virtual Bridge automatically. Data is transferred between the bridges over Ethernet, even though the Ethernet is not shown in this diagram to keep the illustration simple. A typical project with all the connections is shown in Figure 4.

Figure 4: Typical Project



Network Monitoring

The fact that a cable (or 3) can fail before anyone notices any data loss can be a false solution if cable faults go undetected. One approach is to have the bridge kernel monitor the health of all serial and network cables connected to each bridge, and then to present this in a table that can be read from the host application. Alternatively, a web browser, or a special monitor can reside on on the network, receiving error reports. Optionally FieldServers can send SNMP traps to indicate cable faults to a SNMP client.

This article was written by Edward Hague, Chief Technology Officer at FieldServer Technologies

FieldServer Technologies

For further information, please contact
sales@fieldserver.com
Phone (408) 262-2299 (888) 509-1970
Or visit www.fieldserver.com

Reprinted from The Industrial Ethernet Book November 2001