High availability process safety applications enabled by Concurrent Connections

Filip Zembok
Principal Engineer, Embedded Software
Rockwell Automation

Gregory Majcher
Principal Application Engineer, Open Architecture Management
Rockwell Automation

Darren Klug
Project Engineer, Embedded Software
Rockwell Automation

Presented at the ODVA
2023 Industry Conference & 22nd Annual Meeting October
October 18, 2023
El Vendrell, Spain

Abstract

In most process control and many manufacturing applications, control system failure resulting in unexpected shutdown can cause financial loss through wasted products and system restart can take an extended period of time. The ability to design fault-tolerant control systems for these applications is critical. This is even more important when the response to a safety-critical incident requires a highly controlled transition to a safe state, rather than the instant stop of moving equipment normally used in manufacturing applications. These applications rely on the control system being “highly available” even in the face of unexpected failures. Products can be designed to minimize failures, but not eliminate them. Systems can employ redundancy to remove the chance of a single failure rendering the control system inoperable. Both techniques help, but still have limitations.

In the Spring 2023 publication of the CIP family of specifications, ODVA announced the addition of an important new technology, Concurrent Connections which enables flexible, zero switchover time, end-to-end redundancy solutions. This paper provides a brief introduction to how availability is measured, documents some of the issues with current high availability solutions, and highlights how Concurrent Connections address them. It will also summarize the portions of the CIP specifications that were modified to add this new functionality.

Keywords:

Concurrent Connections, High Availability, Redundancy

Overview of Redundancy System Concepts

Availability

Availability in the context of industrial automation systems refers to the ability of the system to perform its intended functions as expected for a specified period of time. Availability is expressed as a percentage of the total operating time of the system. Availability is calculated with the following formula:

Availability = MTTF / (MTTF+MTTR)

where:

MTTF is Mean Time To Failure. MTTF is the average time that a system can provide service before experiencing a failure. MTTF for the system is calculated based on the MTTF of its components. MTTF of a single module is provided by its vendor and is calculated based on the specification of parts used to build a module, the module design, and the number of module warranty returns after one year.

MTTR is Mean Time To Restore/Repair. MTTR is the average time it takes to resume system service after a failure has been experienced. MTTR includes the time it takes to detect the failure. The value of MTTR depends on many factors and is specific to the system, this value is provided by the system constructor.

High Availability

High availability is based on the concept of availability. High Availability is the term used to describe a higher amount of availability for the system than standard availability. High availability is often expressed as a "number of nines". See the table below.

“Number of nines”	Availability %	Possible Downtime per Year
2	99 %	3.65 days
3	99.9 %	8.76 hours
4	99.99 %	52.6 minutes
5	99.999 %	5.26 minutes
6	99.9999 %	30 seconds

High Availability can be achieved by maximizing MTTF and minimizing MTTR. Maximizing MTTF for a single module can be achieved by using high-quality components that have proven reliability and are designed to withstand the specific operating conditions of the industrial automation system. Maximizing the MTTF of a single module has its technological limits. Maximizing the MTTF of the system is achieved by applying redundancy to system components. In order to minimize MTTR, the failure needs to be detected as quickly as possible and repaired as quickly as possible. Low MTTR is achieved by reliable diagnostics, training of the system maintenance staff on the repair procedure, and availability of spare system components.

Redundancy

Redundancy in the context of industrial automation systems refers to the use of duplicated components that are designed to provide backup support and mitigate the failure or reduce the consequences of a failure.

There are multiple aspects of redundancy:

- Backup type: Hot, Warm, or Cold
  - Hot. The backup is completely ready to take over when the active device fails.
  - Warm. The backup is powered up but requires an action to be activated.
  - Cold. The backup is not powered up.
- Synchronization. Active or Passive
  - Active. The backup is kept in a synchronized state with the active device.
  - Passive. The backup is not synchronized with the active device.
- Switchover or Concurrent
  - Switchover. Only the active device is actively participating in the process, the backup device will be activated when the active device fails.
  - Concurrent. The devices that are redundant are functionally equivalent and simultaneously participate in the process, they are all backups of each other.

Fault Tolerance

Fault Tolerance in the context of industrial automation systems is the ability of the system to continue its intended operations in the presence of failures. Redundancy is one way to achieve or increase Fault Tolerance.

The system can be designed to have fault tolerance in a selected part of the system. System parts are:

- Controllers
- Power supplies
- Network Infrastructure
- IO devices
- Field devices

It is up to the system constructor to choose where to apply redundancy and where to achieve fault tolerance. A system that supports fault tolerance for every device within the system is deemed a “no single point of failure” system.

An example of a technology that can tolerate failure in the system is Device Level Ring (DLR) [9]. A system that uses DLR can continue its operations in the presence of a single fault of network media (cable) that connects devices in the ring.

Another example of technology designed for fault tolerance is Parallel Redundancy Protocol (PRP) [8], [9]. A system that uses PRP can continue its operations in the presence of multiple failures of network media so long as one path through the network remains available between participants in the connection.

High Availability in Process Industry

The term “Process Industry” encompasses many different industries. Examples of process industries are:

- Power generation (production of electricity through various methods, such as coal or gas-fired power plants and nuclear power plants)
- Oil and gas (exploration, production, refining, and transportation of oil and gas)
- Mining and minerals (extraction and processing of minerals, such as coal, metals, and industrial minerals)
- Food and beverage (production of food and beverage products, such as baked goods, soft drinks, and processed foods)
- Pharmaceuticals

- Chemical manufacturing
- Pulp and paper (production of paper products, such as newspapers, magazines, and packaging materials)
- Textile manufacturing (production of textiles and clothing, such as cotton, wool, and synthetic fibers)
- Cement manufacturing
- Water treatment (treatment of water to remove impurities and make it safe for consumption or industrial use)
- Semiconductor manufacturing
- Paints and coatings manufacturing
- Glass manufacturing

The common characteristics of process industries are that they involve the production of physical and/or chemical products through a series of continuous or batch processes. Process industries installations are usually large-scale and complex, and they transform high volumes of materials. Processes may involve high temperatures, high pressures, and other hazardous conditions, which can make it difficult, risky, or even impossible to stop the process abruptly. Those processes often involve the use of raw materials, which cannot be easily stored or reused once the process has been stopped. For example, in the oil and gas industry, oil and gas reserves cannot be easily shut down and restarted, and the reservoirs may be damaged if the process is abruptly stopped or restarted. Similarly, in the chemical industry, stopping a reaction process prematurely can result in the loss of valuable raw materials and products. Another example is glass production, when the glass melting process begins it is typically operated continuously for several years to maintain the high temperatures required for the melting process. Stopping such a process quickly can damage the process equipment or the glass product.

The sudden stoppage or the loss of control of an industrial process can have catastrophic consequences, including loss of human lives, environmental contamination, equipment damage, also significant economic implications, such as lost productivity, lost profits, and increased costs associated with restarting the process.

A few examples of catastrophic incidents in the process industry:

- Bhopal gas tragedy [2], 1984, Bhopal, Madhya Pradesh, India. The toxic gas (methyl isocyanate) leak at a Union Carbide India Limited pesticide plant caused the deaths of an estimated 3,000 people immediately, and an additional 15,000 deaths in the years following the incident. Over 500,000 people were injured.
- Piper Alpha oil rig explosion [3], 1988, Piper Alpha oil rig in the North Sea. An explosion resulted in the deaths of 167 workers and caused significant damage to the environment. The total insured loss was about 2 billion US dollars (1988). At the time of the disaster, the platform accounted for approximately 10% of North Sea oil and gas production.
- Deepwater Horizon oil spill [4], 2010, Gulf of Mexico. The oil rig explosion caused the loss of 11 lives and an oil spill that resulted in significant environmental damage.
- Tianjin explosions [5], 2015, Tianjin, China. An explosion at a chemical storage facility caused the deaths of 173 people, injured hundreds more, and caused significant damage to the surrounding area.

Because of the severe consequences of industrial process failure, the process industries are highly regulated. See [6] and [7].

High Availability is critical to the process industry for the following reasons:

- Safety - High Availability can help prevent accidents and protect workers from harm.
- Productivity - High Availability minimizes process downtime.
- Company reputation - The severe consequences of industrial process accidents can have a significant impact on a company's reputation. High Availability can help prevent accidents.
- Cost savings - High Availability can help prevent accidents and thus minimize the need for damaged equipment repairs and replacements.

High Availability and Redundancy in the CIP Specification

The CIP Networks Library currently contains some solutions for high availability and redundancy. These solutions do not cover the whole system and are missing some details, leading vendors to create vendor-specific solutions.

CIP Networks Library, Volume 1 Common Industrial Protocol [10] defines a Redundant Owner connection type that enables multiple Controllers to take ownership of a device's outputs in a standardized way. The solution involves the use of the Redundant Owner bit in Network Connection Parameters of the Forward_Open request and Claim Output Ownership (COO) and Ready for Ownership of Outputs (ROO) bits in the connection real-time header. This solution does not address the redundancy of connection targets. The Redundant Owner and ROO, COO solution for redundancy was popular for ControlNet devices but this popularity did not transfer to EtherNet/IP devices.

CIP Networks Library, Volume 2 EtherNet/IP Adaptation of CIP [9] defines ways to achieve media redundancy using Device Level Ring (DLR) and Parallel Redundancy Protocol (PRP).

CIP Networks Library, Volume 4 ControlNet Adaptation of CIP [11] defines ways to achieve media redundancy and ring topologies.

Existing Redundancy solutions and their problems

Despite limited support for redundancy in the CIP Networks Library, vendor-specific redundancy solutions based on CIP are available in the market. On the one hand, there is a need to standardize the redundancy scheme to support the seamless integration of devices that support redundancy from different vendors. On the other hand, redundancy is used for critical missions, and in order to minimize the risk of failure vendors tend to release "redundancy bundles" that gather a specific set of devices in specific versions, and only guarantee proper redundancy system behavior for the "redundancy bundle".

One example vendor-specific, CIP-based redundancy solution takes advantage of the flexibility of EtherNet/IP networks. The family of TCP, UDP, IP, and Ethernet protocols is used as an abstraction to enable a backup controller to take over the responsibilities of the active controller in the event of active device's failure. The active device has an IP address and MAC address that identify it in an EtherNet/IP network and that are used to communicate with the rest of the system. The transfer of responsibilities to the backup device is known as switchover or failover [1]. During the switchover process, the backup device takes over the IP address of the failed device and sends Gratuitous ARPs to announce that this IP address is now associated with a new MAC address. The backup device continues the work of the active device that has failed as shown in Figure 1.

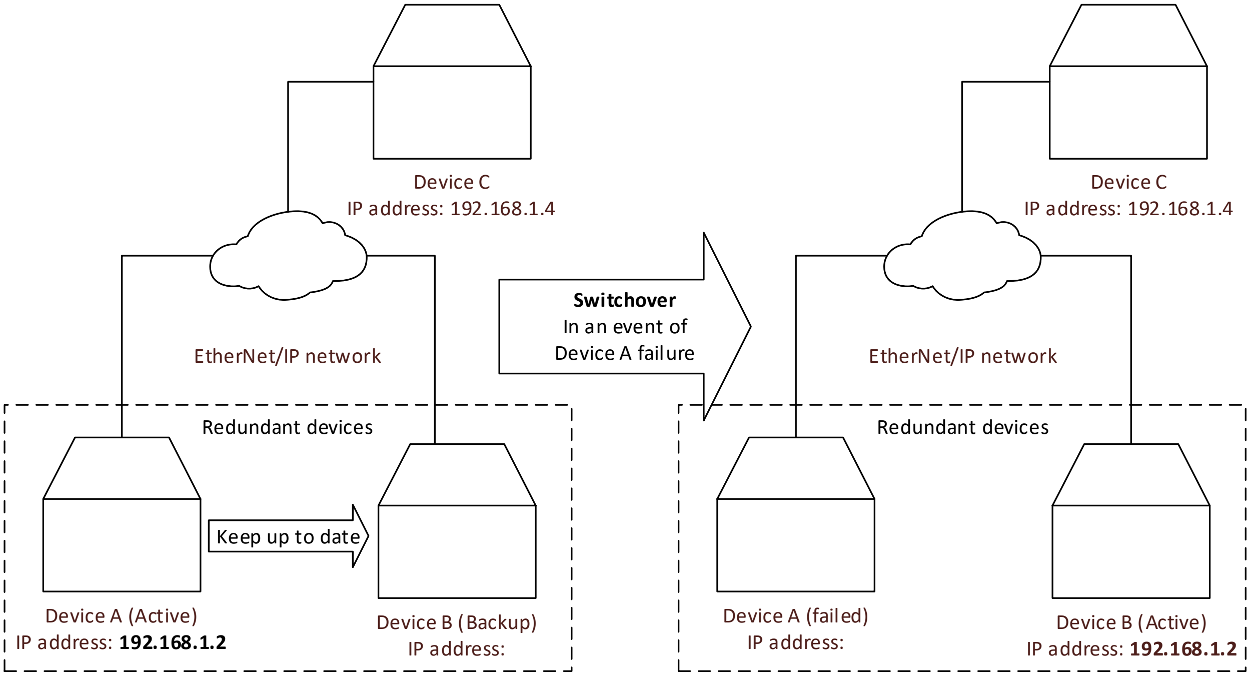


Figure 1

The solution described above is an example of switchover redundancy.

The solution described above has pros and cons. The pros are:

- Not all the devices connected to EtherNet/IP networks need to support redundancy. Parts of the system can even be unaware that redundancy is used in another part of the system. This supports easy redundancy solution integration with any EtherNet/IP capable devices.
- When combined with Hot and Active aspects of redundancy, this solution is capable of maintaining CIP implicit connections if the switchover can be executed before the connection timeout timers of those connections expire.

The cons are:

- During the switchover there is a nonzero period of time when CIP implicit connection data is not sent by any of the redundant devices. The process is not controlled for this period of time.
- The switchover time constrains values of CIP connection parameters. The Requested Packet Interval (RPI) and Connection Timeout Multiplier values must be adjusted, so the connection does not time out during the switchover period. The connection timeout settings are especially important for CIP Safety connections as the timeout of such connection would cause the system to transition into a safe state, and this would be a “spurious trip”. Long connection timeout settings are unacceptable for customers that want both redundancy and quick detection of connection problems.
- The switchover times that can be guaranteed with this technique are too long for some systems. For safety systems, a long switchover time means a long safety reaction time that can impact the physical constraints of a system in order to keep humans and equipment safe. Long safety reaction times can even make the system infeasible if it cannot be out of control for that period of time.
- For every CIP connection maintained by the active device, the O→T Network Connection ID and T→O Network Connection ID need to be passed from the active device to the backup device to enable switchover without dropping CIP connections.
- CIP Explicit connections are TCP-based in EtherNet/IP and synchronizing TCP sessions between the active device and the backup devices is challenging, thus usually CIP Explicit connections are

dropped at switchover. This leads to a loss of communication with HMI devices and the need for reconnection.

- This redundancy technique only applies to EtherNet/IP originators.

Another approach that some targets have used is to delay applying their connection fault action for a configurable amount of time after a connection times out. If another (or even the same) originator re-establishes the connection before this additional delay expires, the connection fault action is avoided. The pause in control during the additional delay required for this approach is not acceptable for all applications.

Concurrent Connections to the rescue

The spring 2023 publication of the CIP Networks Library introduced Concurrent Connections technology. Concurrent Connections are an answer to many of the redundancy solution deficiencies identified above. In short, Concurrent Connections enable flexible, zero switchover time, end-to-end redundancy solutions. The easiest way to explain Concurrent Connections is to compare them to PRP. Concurrent Connections are like PRP on the CIP connection layer. Concurrent Connections allow the use of redundancy for any endpoints or routers along the path of the connection. Concurrent Connections enable multiple paths for transferring the CIP data between all participants in the connection. The CIP data is sent simultaneously across multiple branches through all of the participants to reach the other end of a connection as shown in Figure 2..

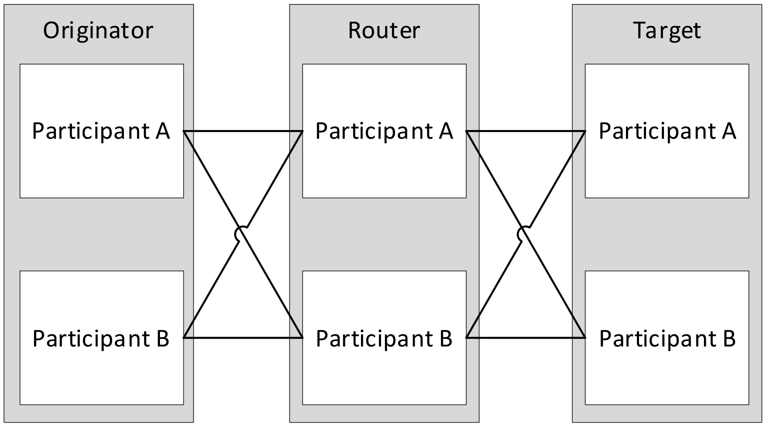


Figure 2 Example system with concurrent connection

There are three redundant devices in the above diagram: a duplex originator, a duplex router, and a duplex target. The concurrent connection is represented as all the links between all participants in the above diagram. There are 8 such links, the CIP specification calls them branches of the concurrent connection. Because of this concurrent connection topology, there are multiple paths that data from either of the originator participants can use to reach either of the target participants and vice versa. All of the concurrent connection branches together form one logical CIP Application Connection. From the application perspective, the concurrent connection looks exactly the same as the non-concurrent connection.

If a device participating in the concurrent connection fails, then its concurrent connection branches will stop working. When the network path used by the concurrent connection fails, the affected concurrent connection branches will stop working. The remaining branches of the concurrent connection will keep on delivering data between originators and targets as long as there is at least one available path between originators and targets. Figure 3 demonstrates the setup from Figure 2 after a few failures.

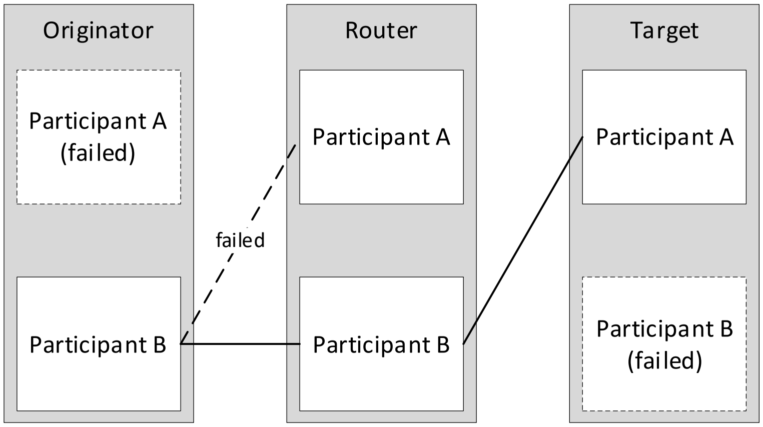


Figure 3 Example system with concurrent connection, with device and network path failures

Originator Participant A and Target Participant B have failed and are unavailable. There are no concurrent connection branches from those devices. Also, the network path between Originator Participant B and Router Participant A failed. The concurrent connection branches between Router Participant A and targets have timed out as Router Participant A is not able to route toward originators. Despite multiple failures, the concurrent connection is still able to exchange CIP data between an originator and a target. From the application perspective, the connection is working, and the process can be continued.

Concurrent Connections flexibility

Concurrent Connections are a flexible solution as they enable CIP device redundancy at every participant along the connection path. It is up to the system designers to decide where to use redundancy and what depth of redundancy (duplex, triplex, etc.) shall be applied. For example, Figure 4 presents a system that focuses on Controller redundancy.

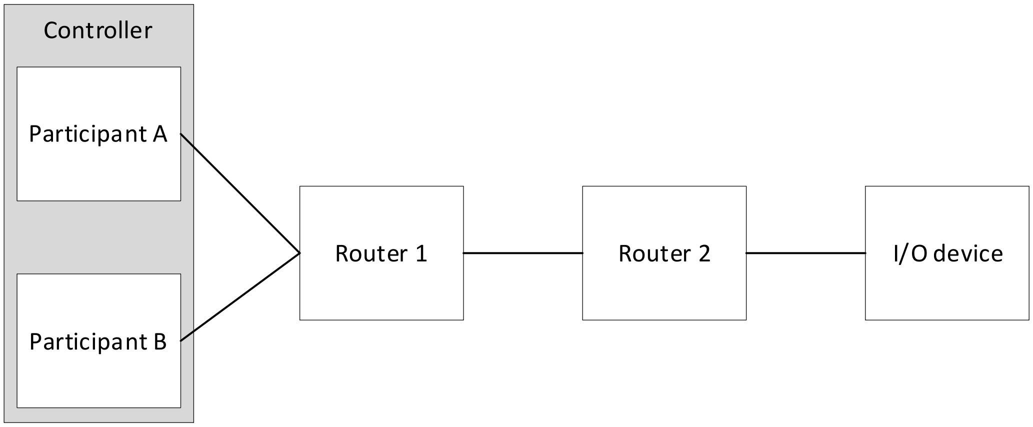


Figure 4 Example concurrent connection topology 1: Controller redundancy

Figure 5 presents a system that focuses on I/O device redundancy.

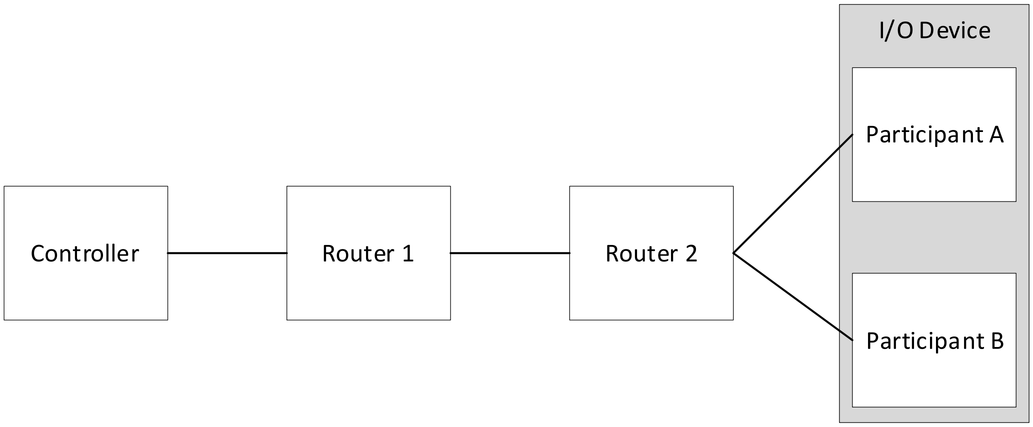


Figure 5 Example concurrent connection topology 2: I/O device redundancy

Figure 6 presents a system that focuses on network adapter redundancy.

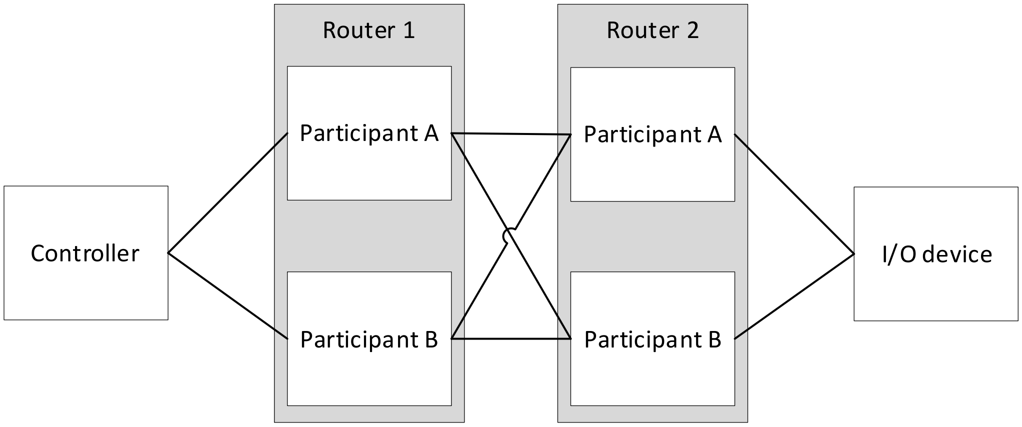


Figure 6 Example concurrent connection topology 3: Network adapter redundancy

Concurrent Connections technology does not limit the number of devices that realize a certain function of the system. Figure 7 demonstrates a system of triplex IO devices.

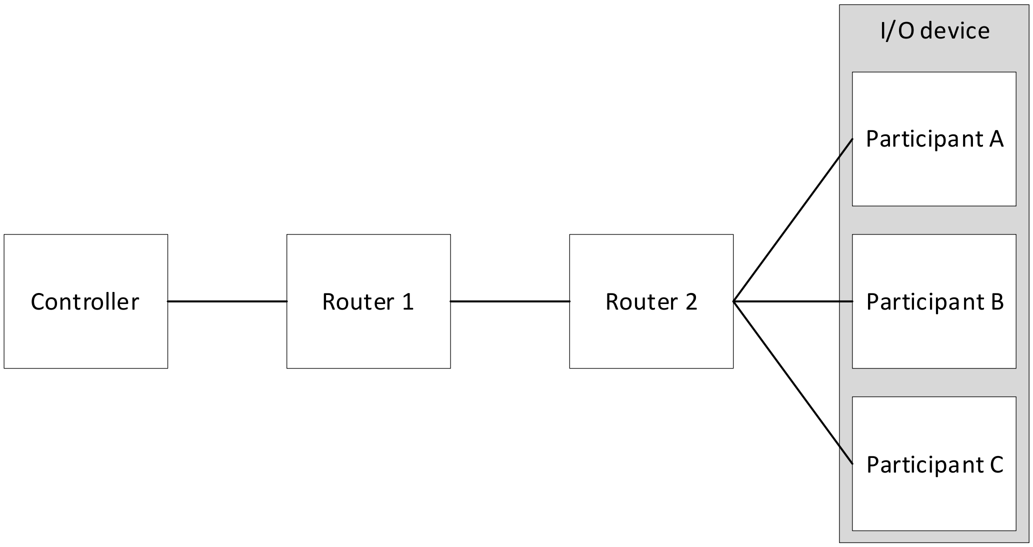


Figure 7 Example concurrent connection topology 4: Triplex I/O devices

Concurrent Connections zero-switchover time

Concurrent Connections enable hot, active, and concurrent redundancy. In the Concurrent Connections solution, the redundant devices that realize the part of the system are functionally equivalent and they are all backups to each other. The duplicated devices are kept in a synchronized state to the point that allows all of them to participate in the control process. The synchronization mechanism is vendor specific. The redundant devices on the endpoints of concurrent connections (originators and targets) are synchronized to send the same CIP connection payload with the same Concurrent Connection Sequence Count (CCSC) at the same time. This synchronization of the CIP connection data and the production time does not need to be strict; deviations are acceptable as long as they are within the boundaries defined within the Concurrent Connections definition in Volume 1. Figure 8 depicts synchronization between redundant Originators and Targets.

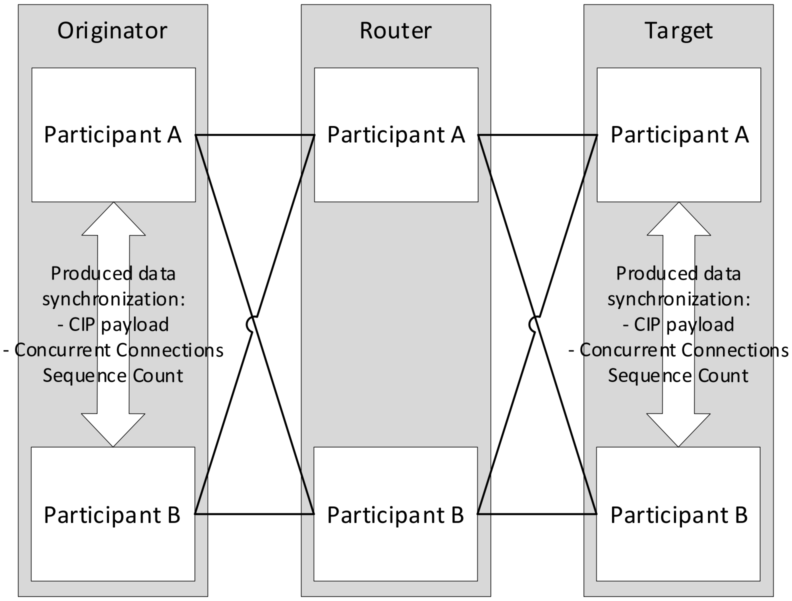

Originator
Router
Target
Participant A
Participant A
Participant A
Produced data synchronization:
- CIP payload
- Concurrent Connections Sequence Count
Produced data synchronization:
- CIP payload
- Concurrent Connections Sequence Count
Participant B
Participant B
Participant B

Figure 8 Synchronization of CIP connection payload between redundant endpoints

Concurrent connection data is sent via all concurrent connection branches and there are multiple paths between connection endpoints. Concurrent Connections use the CCSC in their runtime header to deal with packet duplicates. The first packet with a given CCSC value is forwarded by routers and consumed by endpoints, subsequent packets with the same CCSC value are dropped. When one of the duplicated devices fails the CIP connections are kept alive by its remaining partners, and the process can be continued seamlessly. The Concurrent Connections solution eliminates the switchover and thus mitigates its deficiencies.

Concurrent Connections branch recovery

When the concurrent connection branch fails, and the concurrent connection can continue CIP data delivery via other branches, the device that detected the local failure of the concurrent connection branch starts the procedure of branch recovery. The dashed lines in Figure 9 depict branch recovery on the branches connected to the failed device.

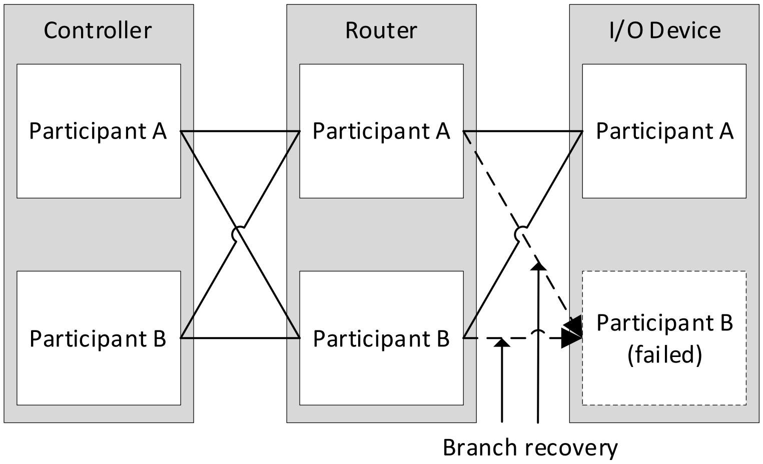


Figure 9 Branch recovery

When devices detect a timeout on the concurrent connection branch towards I/O Device Participant B, those devices start to periodically resend connection open requests to I/O Device Participant B. The attempts to reopen the concurrent connection branches continue until I/O Device Participant B responds with a successful connection open response.

The branch recovery procedure is local; in this case, only Router Participant A and Router Participant B notice the local failures of the concurrent connection. Those routers also report an issue via Concurrent Connections diagnostics, so the maintenance staff easily localize the problem and can start repair actions. In this case, Controller Participant A and Controller Participant B do not notice the local failures of the concurrent connection.

Concurrent Connections vs existing redundancy solutions.

Concurrent Connections do not solve all the problems of systems that require high availability, but they are an important building block of such systems. Concurrent Connections standardize the communication protocol that enables high availability systems.

Compared to the existing redundancy solutions described earlier in this paper, Concurrent Connections offer:

- A high-level CIP protocol solution that is independent of network technologies used to connect devices participating in the connection.
- One standardized end-to-end solution for redundant device communication across a system with devices from multiple vendors.
- Flexibility. The redundant devices can be used in any part of the system independently of other parts of the system. There can be a system that uses redundancy only in one of its parts, and there can be a system that applies redundancy in all its parts. The Concurrent Connection protocol does not limit the number of devices that are duplicated in the system part.
- Elimination of switchover and its deficiencies. In the Concurrent Connections solution, the redundant devices participate in the control process all the time. In the event of failure of one of the redundant devices, the remaining devices maintain the CIP connections. There is no system downtime, the MTTF is maximized. The system maintainers have time to replace the failed device.
- Possibility of building a system with no single point of failure.
- Active recovery of local concurrent connection branches. This enables a “plug and play” experience when replacing the faulted device.
- Ease of extension. The Concurrent Connections protocol (new Connection Management services for managing concurrent connections and Concurrent Connection Header) has built-in versioning that can be used to mitigate breaking changes in case of protocol extensions.

Redundancy solutions come with a price and Concurrent Connections are no exception:

- Within a concurrent connection, the packets with the same CIP payload are sent through all concurrent connection branches. For advanced concurrent connection topologies, this leads to higher use of network bandwidth and higher use of CIP devices' processing power.
- Concurrent Connections require active synchronization of the redundant endpoints of the concurrent connection. This aspect is not standardized in the CIP specification since it is anticipated that the redundant participants will all be the same device from the same vendor.

Considering the pros and cons listed above Concurrent Connections are the best communication protocol to be used in a system that requires high availability.

Summary of Changes to CIP Specification

As mentioned above, Concurrent Connections enable any number (N_x) of redundant participants to be used at each router or endpoint (originator or target) along the path of a connection, as shown in the following figure:

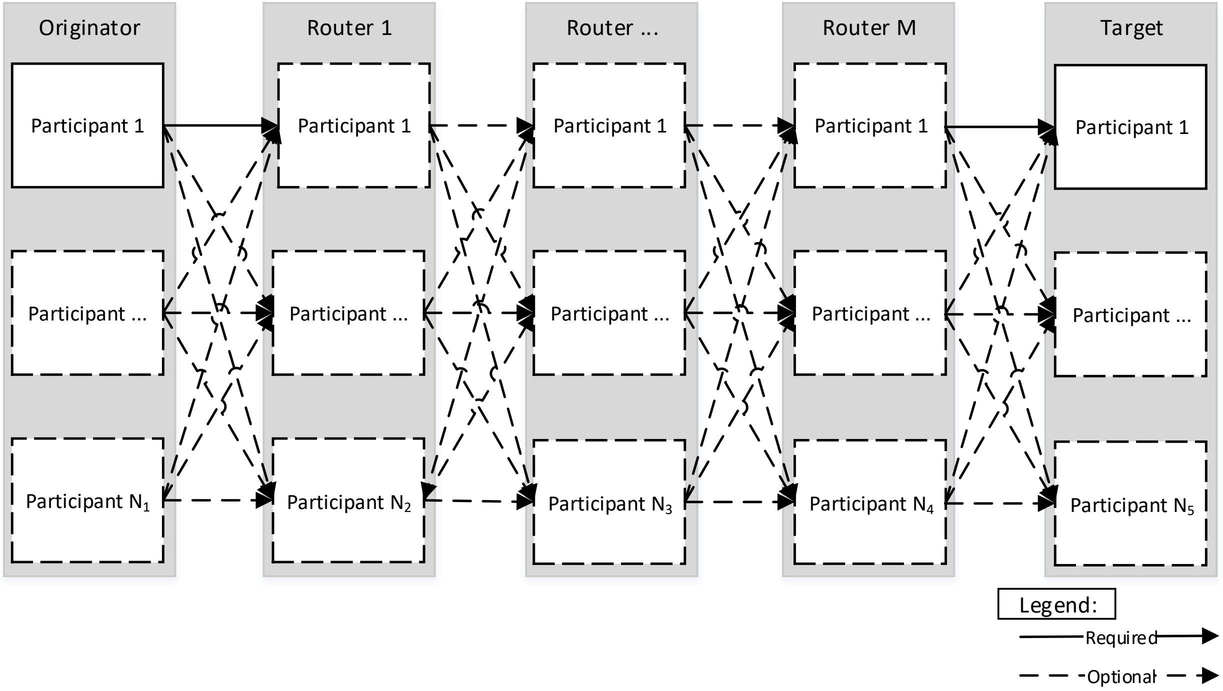


Figure 10 Flexibility of Concurrent Connection topology

The number of participants used for each router or endpoint is determined by the user to achieve the depth of resiliency their application demands.

To manage all of the (redundant) branches between the connected devices, new "Concurrent" Forward Open and Close services are introduced in Revision 2 of the Connection Manager object.

The concurrent open and close services differ from the existing open and close services as follows:

1. The existing Port segments are replaced with a Concurrent Connection Path segment that enables the encoding of multiple concurrent connection branches.

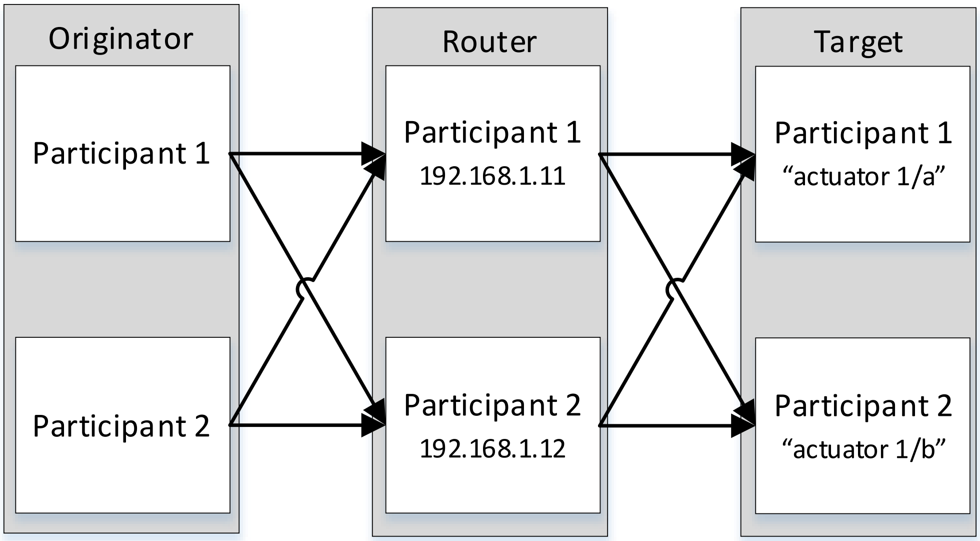


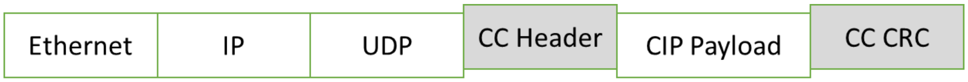
Figure 11 Example system with concurrent connection

2. The Concurrent Connection Path segment for the connection from Figure 11 as configured in the originator participants would contain the following:

Value	Description	
0x5F	Concurrent Connection Path Identifier	
22	Number of 16-bit words that follow	
0x0002	Subtype	
2	Number of hops	
40	Number of bytes in Concurrent Connection Paths that follow	
2	Hop 1	Egress port on Originator Participants 1 and 2
0x12		2 IPv4 addresses follow
0xC0A8010B		IP address of Router Participant 1
0xC0A8010C		IP address of Router Participant 2
3	Hop 2	Egress port on Router Participants 1 and 2
0x22		2 hostnames follow
“actuator 1/a”		Hostname of Target Participant 1
“actuator 1/b”		Hostname of Target Participant 2

3. A Concurrent Connection Protocol Version is introduced to enable future enhancements.

A new Concurrent Connection Packet format is introduced to wrap the existing Real Time formats (e.g. 32-bit header w/ run/idle). The Concurrent Connection Packet format consists of:



1. The Concurrent Connection (CC) Header contains:
   a. Packet Type & Keep Alive field,
   b. Packet Length, and

c. Concurrent Connection Sequence Count

Concurrent Connections are only supported for Transport Class 0 and 1..

Each router participant:

1. Sets up bindings between all of the branches on each side of the participant.
2. Retains the Concurrent Forward Open to automatically resend if a timeout occurs for any of the branches leading to the target.
3. Forwards only the first data packet received for each new data production to all of its branches leading to the next router/endpoint. This applies in both the originator-to-target and target-to-originator directions.

Additional changes:

1. New revisions of the Link Producer, Link Consumer and Connection objects (enhanced to support arrays of Connection IDs – one for each branch)
2. New Connection Manager CC-specific error codes
3. New Connection Manager diagnostic attributes and connection point

Concurrent Connections implementation

As summarized in the preceding section, the Concurrent Connections solution emerges from existing CIP connections.

Concurrent Connections can be understood as a layer added over existing CIP connections, a layer that enables multiple paths between connection endpoints. Concurrent Connections are managed with a new set of Connection Manager services: Concurrent_Forward_Open, Large_Concurrent_Forward_Open, and Concurrent_Forward_Close. Those services are derived from the well-known Forward_Open, Large_Forward_Open, and Forward_Close.

The mechanics of transferring CIP data on a single branch of the concurrent connection is exactly the same as the mechanics used to transfer data in a segment of a non-concurrent connection.

Considering the CIP device that already supports non-concurrent class 0 and 1 connections, the following elements need to be implemented to add a minimum Concurrent Connections capability:

- Concurrent Connection Path Extended Network Segment
- Concurrent_Forward_Open Connection Manager service (the format is the Forward_Open plus one additional field, the Concurrent Connections Protocol Version)
- Concurrent_Forward_Close Connection Manager service (the same format as Forward_Close)
- Counting of concurrent connection branches and connection management decisions based on that count
- Concurrent Connection Header
- Sending packets with Concurrent Connection Header on all open branches
- Receiving packets with Concurrent Connection Header from multiple branches and discarding duplicated packets
- Branch recovery procedure (each participant needs to remember the original Concurrent_Forward_Open and resend it to perform branch recovery)

Additionally:

- Originator participants need to synchronize the parameters of concurrent connections they originate.
- Endpoint participants need to synchronize CIP data, Concurrent Connections Sequence Count, and the production time of packets to be produced. The mechanisms of synchronization between redundant participants are vendor specific.

Support for Concurrent Connections will be available in version 4.2 of the Wireshark network protocol analyzer.

References

[1] A Framework for Implementing Process Applications using CIP Technologies, ODVA 2022 Industry Conference

[2] https://en.wikipedia.org/wiki/Bhopal_disaster

[3] https://en.wikipedia.org/wiki/Piper_Alpha

[4] https://en.wikipedia.org/wiki/Deepwater_Horizon_oil_spill

[5] https://en.wikipedia.org/wiki/2015_Tianjin_explosions

[6] IEC 61511; https://webstore.iec.ch/publication/5527

[7] IEC 61508; Main Safety Standard

[8] IEC 62439-3; PRP Standard

[9] THE CIP NETWORKS LIBRARY, Volume 2, EtherNet/IP Adaptation of CIP

[10] THE CIP NETWORKS LIBRARY, Volume 1, Common Industrial Protocol (CIP™)

[11] THE CIP NETWORKS LIBRARY, Volume 4, ControlNet Adaptation of CIP

****

The ideas, opinions, and recommendations expressed herein are intended to describe concepts of the author(s) for the possible use of ODVA technologies and do not reflect the ideas, opinions, and recommendation of ODVA per se. Because ODVA technologies may be applied in many diverse situations and in conjunction with products and systems from multiple vendors, the reader and those responsible for specifying ODVA networks must determine for themselves the suitability and the suitability of ideas, opinions, and recommendations expressed herein for intended use.