High availability process safety applications enabled by Concurrent Connections

Filip Zembok, Rockwell Automation
Gregory Majcher, Rockwell Automation
Darren Klug, Rockwell Automation

Abstract

In the Spring 2023 publication of the CIP family of specifications, ODVA announced the addition of an important new technology, **Concurrent Connections**. This technology enables **flexible, zero switchover time, end-to-end redundancy** solutions.

This presentation briefly introduces the basic terms and concepts related with **high availability**.

This presentation discusses characteristics of **process industry** systems and points out some of the deficiencies of current high availability solutions.

Finally, the presentation explains the Concurrent Connections technology and describes how **Concurrent Connections improve high availability and safety systems**.

Overview of High Availability

- Availability
  - Mean Time To Failure
  - Mean Time To Restore/Repair
- High Availability
- Redundancy
  - Hot, Warm, or Cold
  - Active or Passive synchronization
  - Switchover or Concurrent
- Fault Tolerance

Availability = MTTF / (MTTF+MTTR)

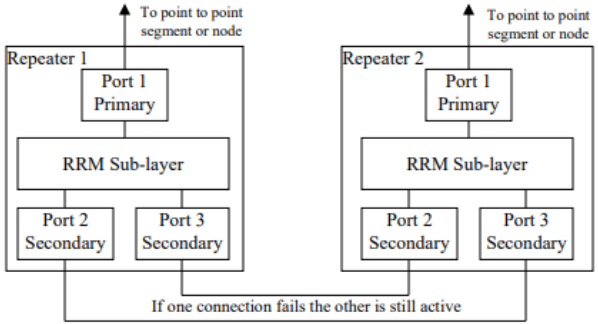
"Number of nines"	Availability %	Possible Downtime per Year
2	99 %	3.65 days
3	99.9 %	8.76 hours
4	99.99 %	52.6 minutes
5	99.999 %	5.26 minutes
6	99.9999 %	30 seconds

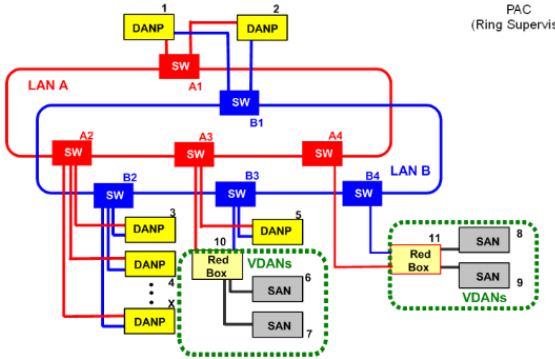
Process Industry

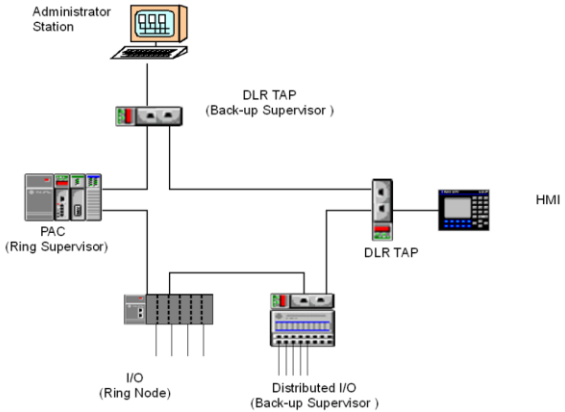
- Large-scale and complex installations
- Processes cannot be easily stopped
- Consequences of failure can be catastrophic
  - Bhopal gas tragedy
  - Piper Alpha oil rig explosion
  - Deepwater Horizon oil spill
  - Tianjin explosions
- High Availability is critical
- Highly regulated

High Availability and Redundancy in the CIP Specification

- CIP Networks Library, Volume 1 Common Industrial Protocol
  - Redundant Owner connection type, Ownership of Outputs (ROO) and Claim Output Ownership (COO) bits
- CIP Networks Library, Volume 2 EtherNet/IP Adaptation of CIP
  - PRP
  - DLR
- CIP Networks Library, Volume 4 ControlNet Adaptation of CIP
  - Media redundancy with ring topologies

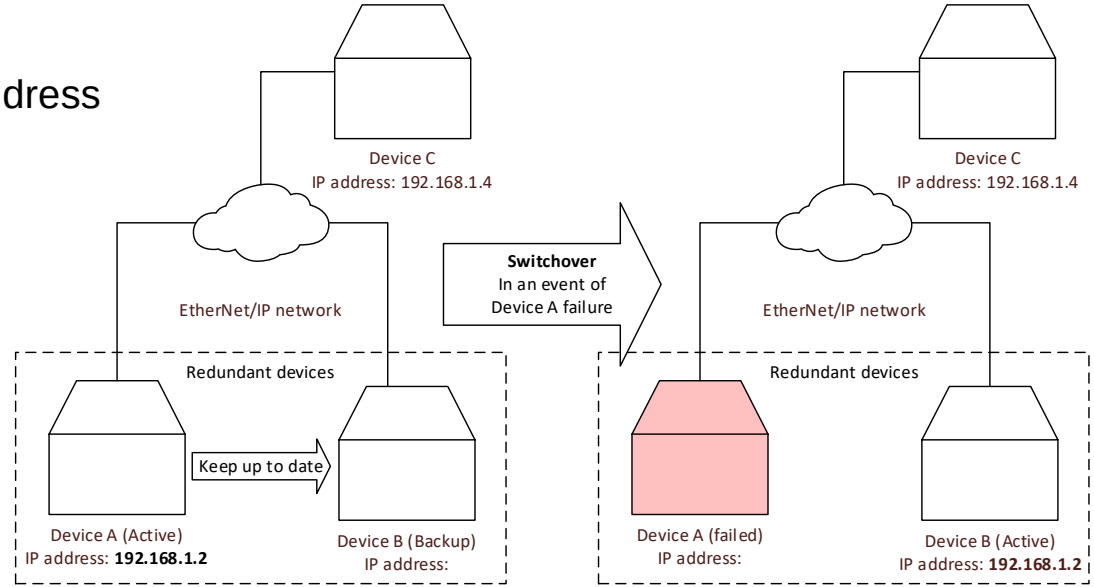






Existing Redundancy solutions and their problems

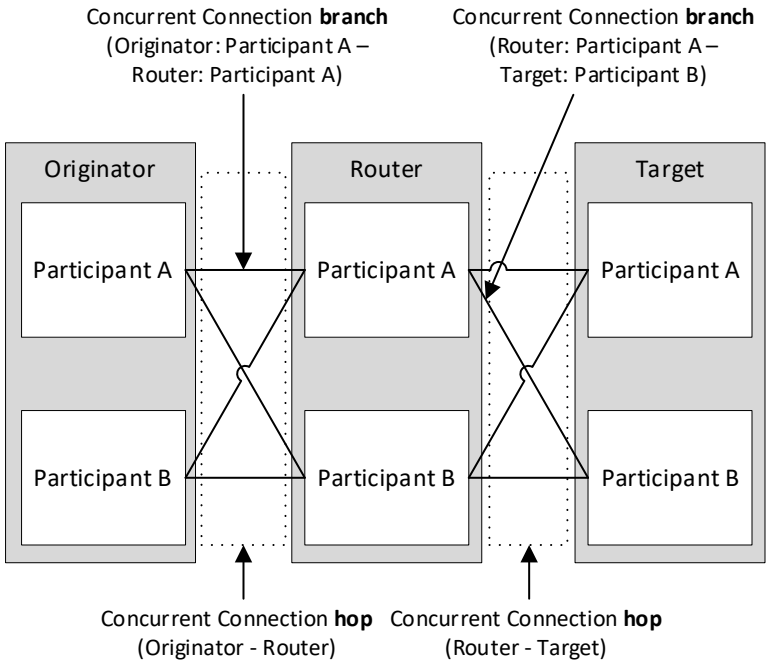
- Vendor specific
- Solution 1: switchover on IP address



- Solution 2: Delay applying connection fault action
- Solution 3: Redundancy realized at the application level
- …

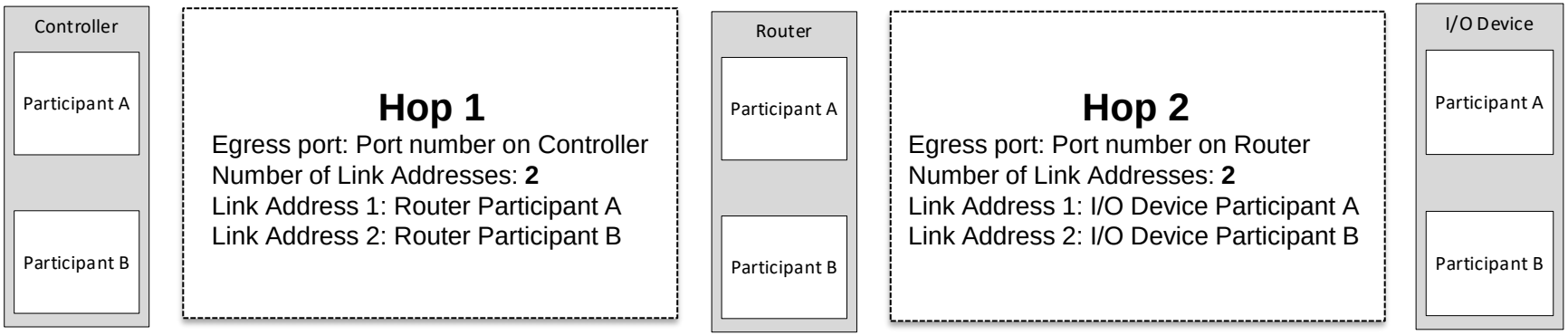
Concurrent Connections

- “PRP on the CIP connection level”
- Redundant participants
- Multiple paths for transferring CIP data
- One logical CIP connection from an application perspective

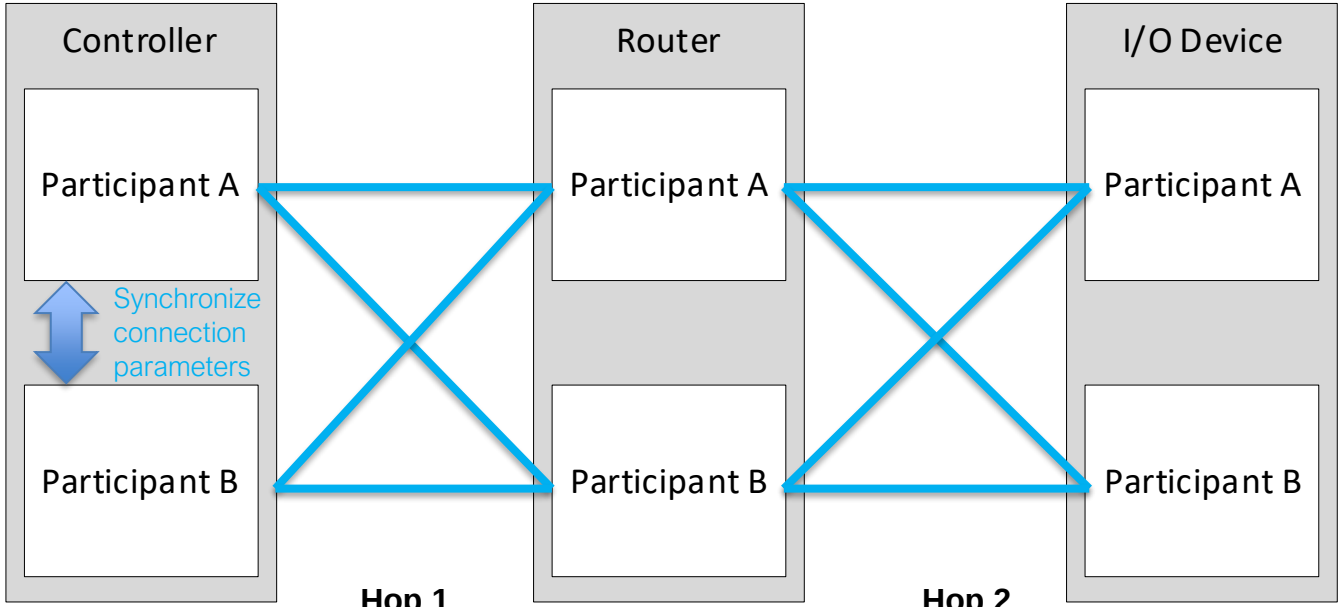


Concurrent Connections – connection management

- New Connection Manager services:
  - **Concurrent_Forward_Open** and **Large_Concurrent_Forward_Open**
    - "Forward_Open + Concurrent Connections Protocol Version field"
  - **Concurrent_Forward_Close**
    - Same format as Forward_Close
- New Extended Network Segment
  - **Concurrent Connection Path** (list of Concurrent Connection **Hops**)



Concurrent Connections - Connection Opening Process



Hop 1

Egress port: Port number on Controller
Number of Link Addresses: **2**
Link Address 1: Router Participant A
Link Address 2: Router Participant B

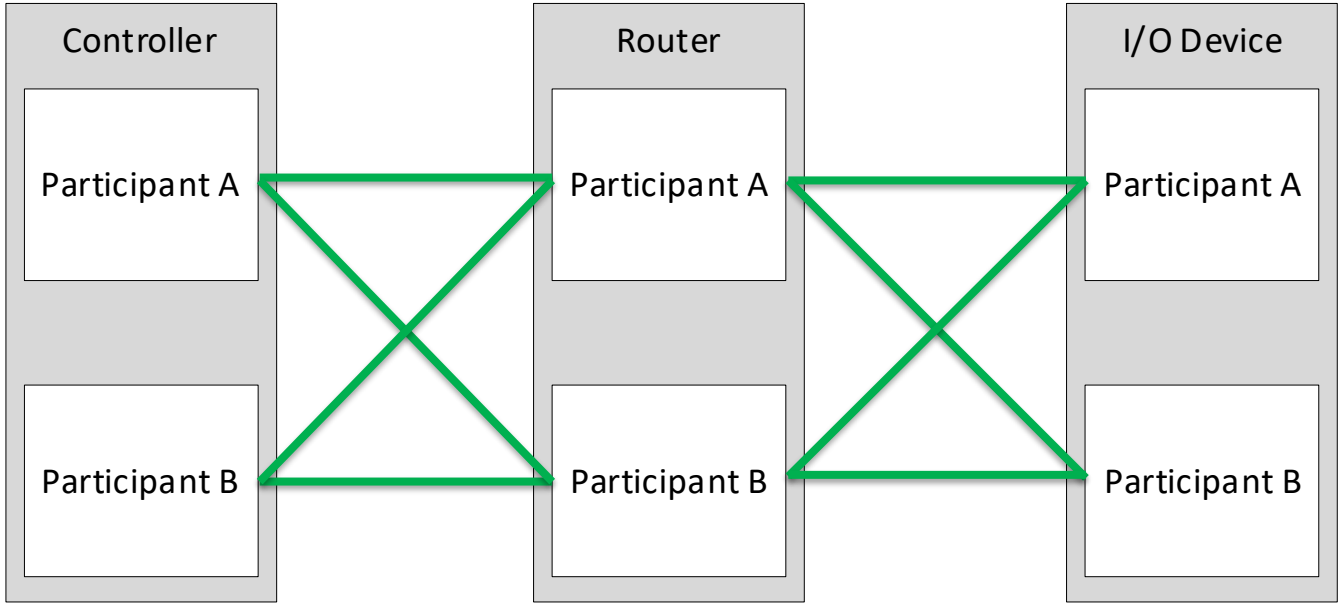
Hop 2

Egress port: Port number on Router
Number of Link Addresses: **2**
Link Address 1: I/O Device Participant A
Link Address 2: I/O Device Participant B

Legend:

- Concurrent Forward **Open request**
- Concurrent Forward **Open response**
- Open branch of Concurrent Connection

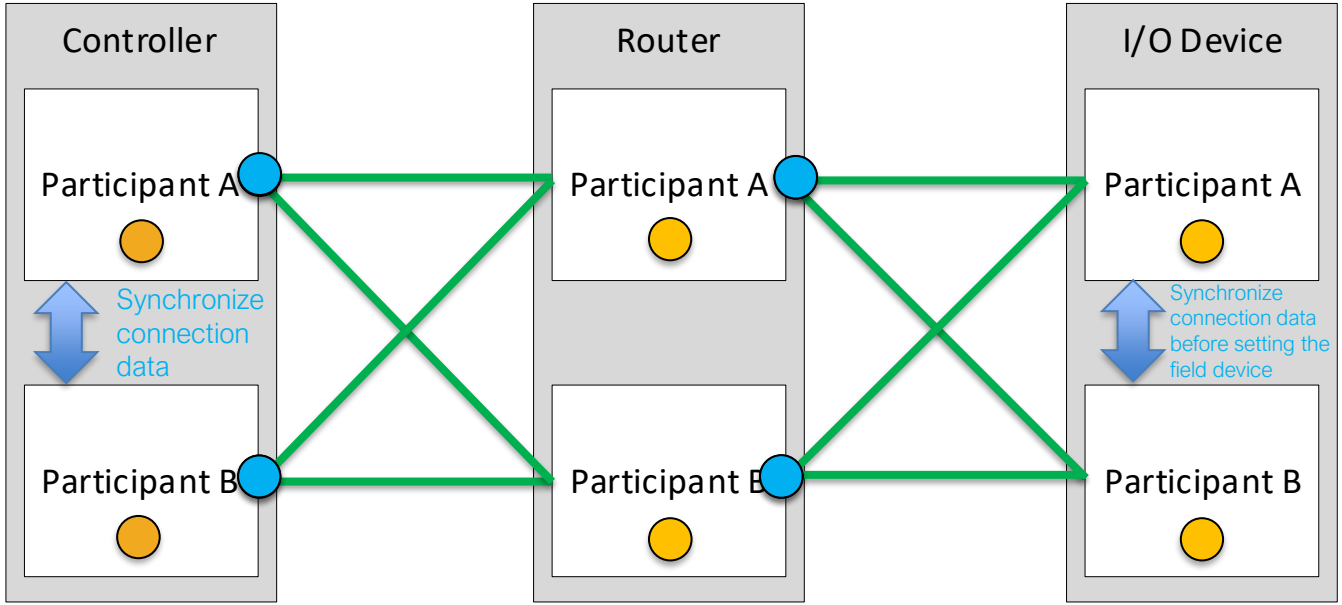
Concurrent Connections - Connection Closing Process



Legend:

- Open branch of Concurrent Connection
- Concurrent Forward Close request
- Concurrent Forward Close response

Concurrent Connections – Data flow

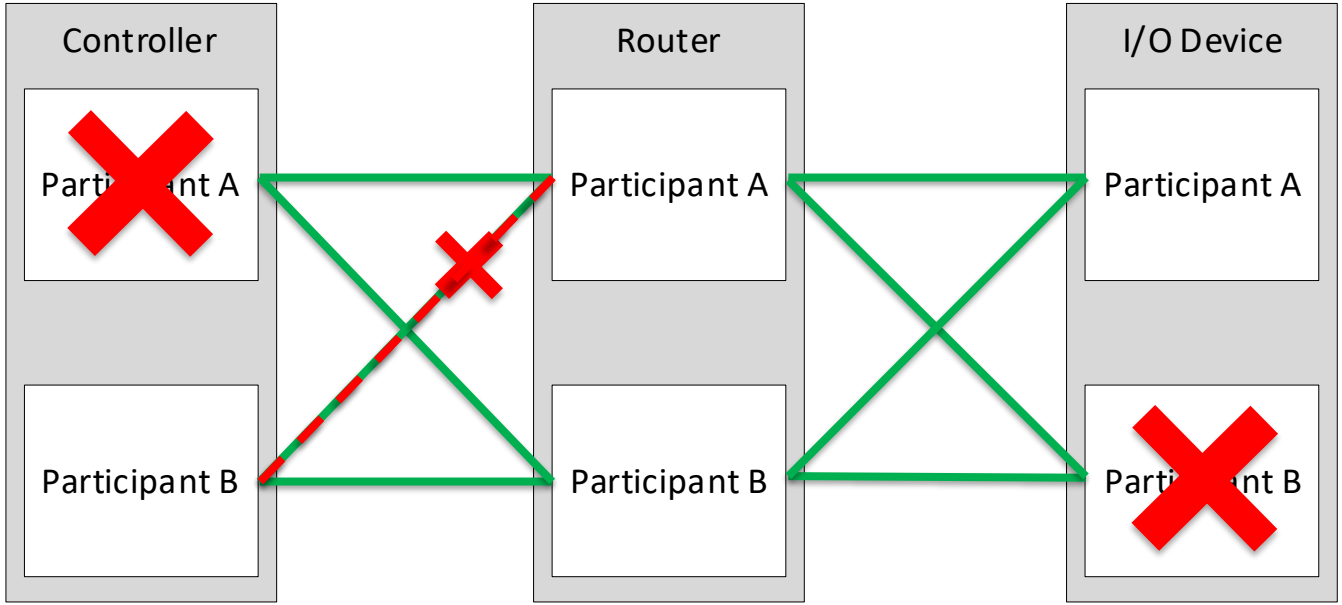


Legend:

- Open branch of Concurrent Connection
- Last Concurrent Connection Packet (CCSC=1)
- Fresh Concurrent Connection Packet (CCSC=2)
- Discarding the redundant packet

The same process happens in I/O Device to Controller direction.

Concurrent Connections - Fault tolerance

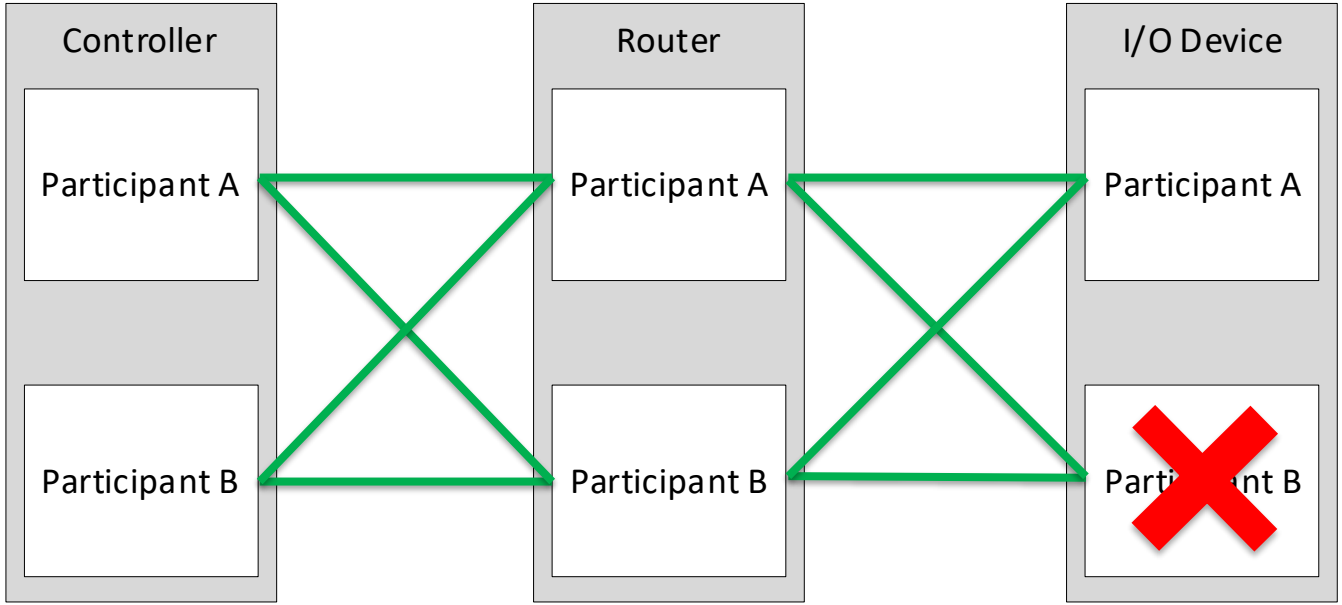


Legend:
- Open branch of Concurrent Connection
- Failure

- The connection remains open as long as there is at least one available path between connection endpoints
- **Zero switchover time**

- Redundant devices are functionally equivalent and participate in the process all the time
- Hot, active, and concurrent redundancy

Concurrent Connections – Branch recovery

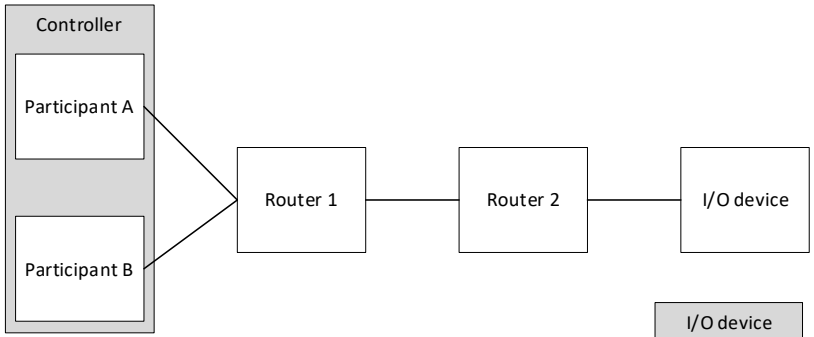


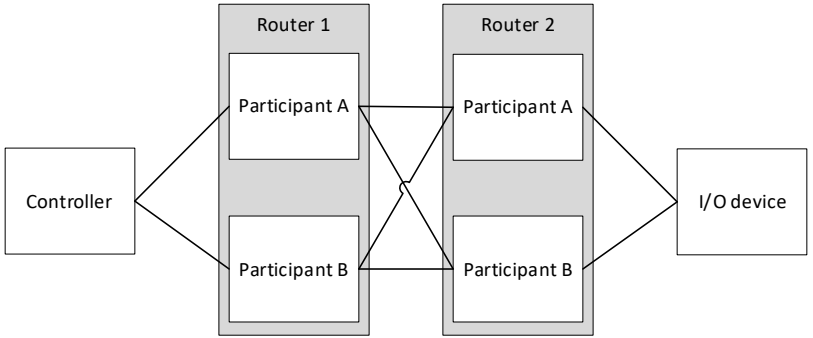
Legend:

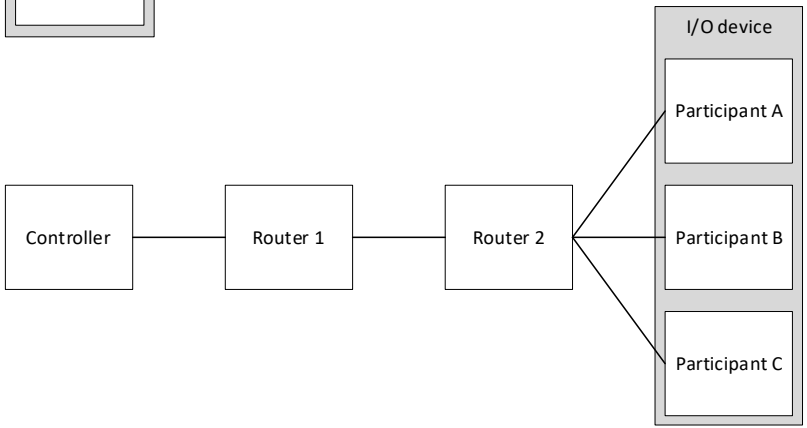
- Open branch of Concurrent Connection
- Failure
- Concurrent Forward Open request
- Concurrent Forward Open response

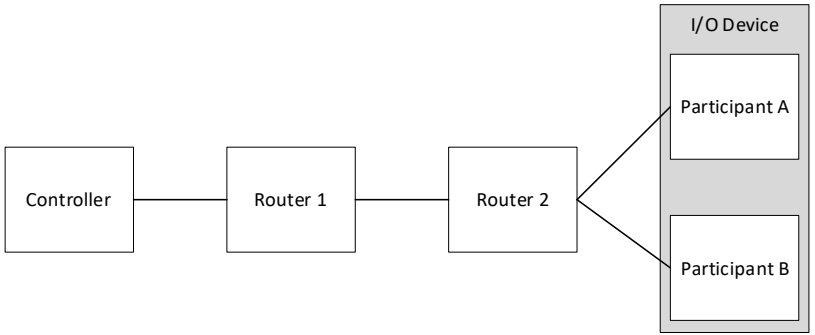
- Local, periodic attempts to reopen failed Concurrent Connection branches

Concurrent Connections - Flexibility



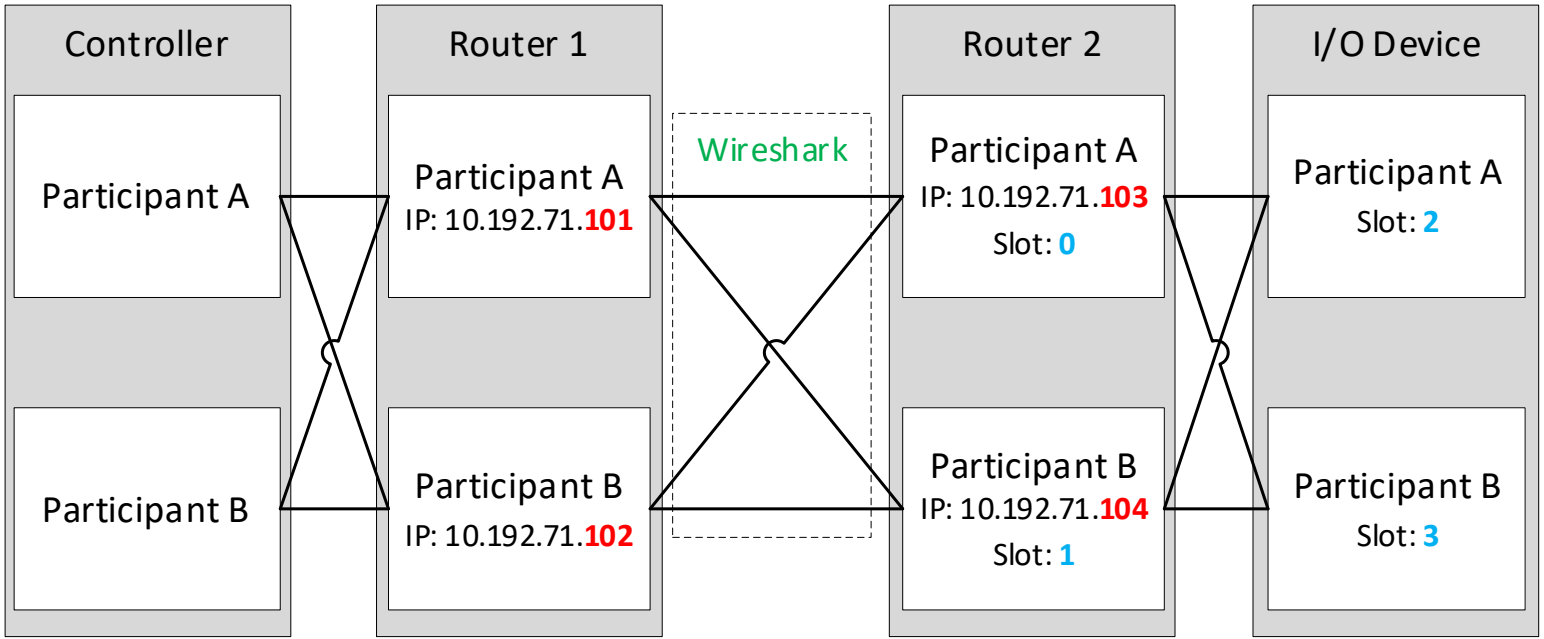






Concurrent Connections - real example

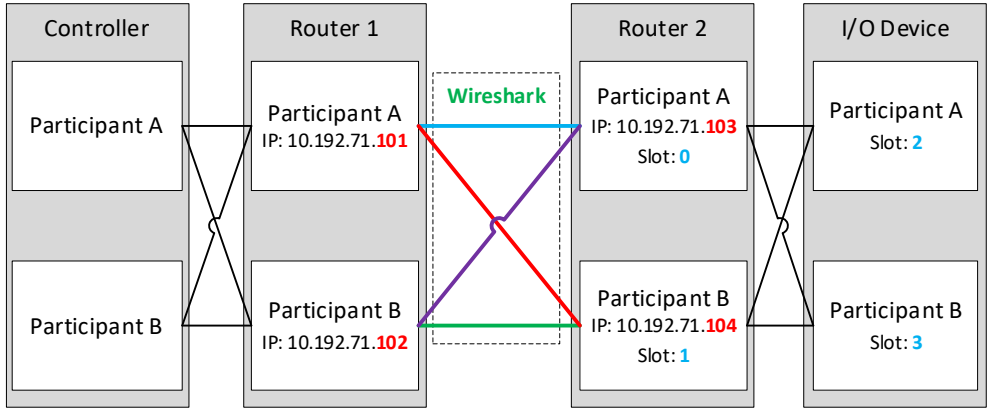
Concurrent Connections will be supported by **Wireshark 4.2** (November 2023 release)



Real example – Concurrent_Forward_Open

cip.cm.conn_serial_num == 0xcccc

No.	Time	Source	Destination	Protocol	Info
2320	0.966484	10.192.71.101	10.192.71.103	CIP CM	Connection Manager - Concurrent Forward Open
2321	0.966579	10.192.71.101	10.192.71.104	CIP CM	Connection Manager - Concurrent Forward Open
2329	0.982349	10.192.71.102	10.192.71.104	CIP CM	Connection Manager - Concurrent Forward Open
2331	0.984646	10.192.71.104	10.192.71.101	CIP CM	Success: Connection Manager - Concurrent Forward Open
2334	0.984744	10.192.71.104	10.192.71.102	CIP CM	Success: Connection Manager - Concurrent Forward Open
2347	0.982268	10.192.71.102	10.192.71.103	CIP CM	Connection Manager - Concurrent Forward Open
2354	0.984334	10.192.71.103	10.192.71.101	CIP CM	Success: Connection Manager - Concurrent Forward Open
2355	0.984431	10.192.71.103	10.192.71.102	CIP CM	Success: Connection Manager - Concurrent Forward Open



Real example – Concurrent_Forward_Open format

```
> Internet Protocol Version 4, Src: 10.192.71.101, Dst: 10.192.71.103
> Transmission Control Protocol, Src Port: 57386, Dst Port: 44818, Seq: 101, Ack: 91, Len: 104
> EtherNet/IP (Industrial Protocol), Session: 0x4000000E, Send RR Data
> Common Industrial Protocol
v CIP Connection Manager
  > Service: Concurrent Forward Open (Request)
  v Command Specific Data
      ...0 .... = Priority: 0
      .... 0100 = Tick time: 4
      Time-out ticks: 186
      Actual Time Out: 2976ms
      O->T Network Connection ID: 0x00000000
      T->O Network Connection ID: 0x0020401a
      Connection Serial Number: 0xcccc
      Originator Vendor ID: Rockwell Automation/Allen-Bradley (0x0001)
      Originator Serial Number: 0x4148384c
      Concurrent Connections Protocol Version: 1
      Connection Timeout Multiplier: *16 (2)
      Reserved: 0x000000
      O->T RPI: 100.000ms
    > O->T Network Connection Parameters: 0x440e
      T->O RPI: 100.000ms
    > T->O Network Connection Parameters: 0x4422
    > Transport Type/Trigger: 0x81, Direction: Server, Trigger: Cyclic, Class: 1
      Connection Path Size: 10 words
    > Connection Path: [Network], Assembly, Instance: 0xBF, Connection Point: 0xC7, Connection Point: 0x0434
    > [Connection Information]
```

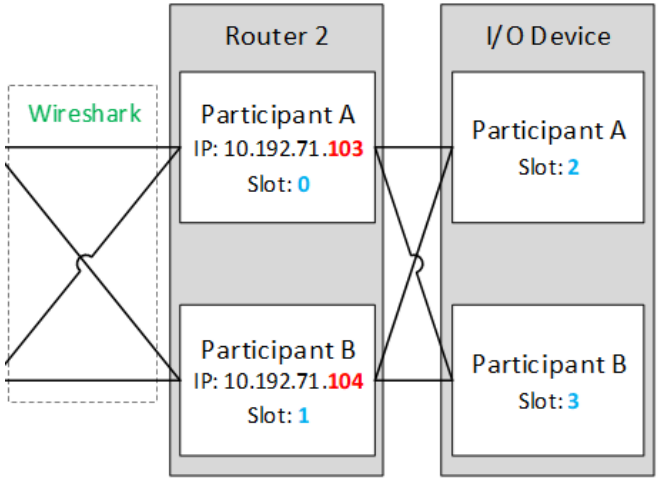
Legend:

- Originators shall use the same values
- Routers and Targets shall check if values are the same for a given Connection Triad
- Difference from Forward_Open

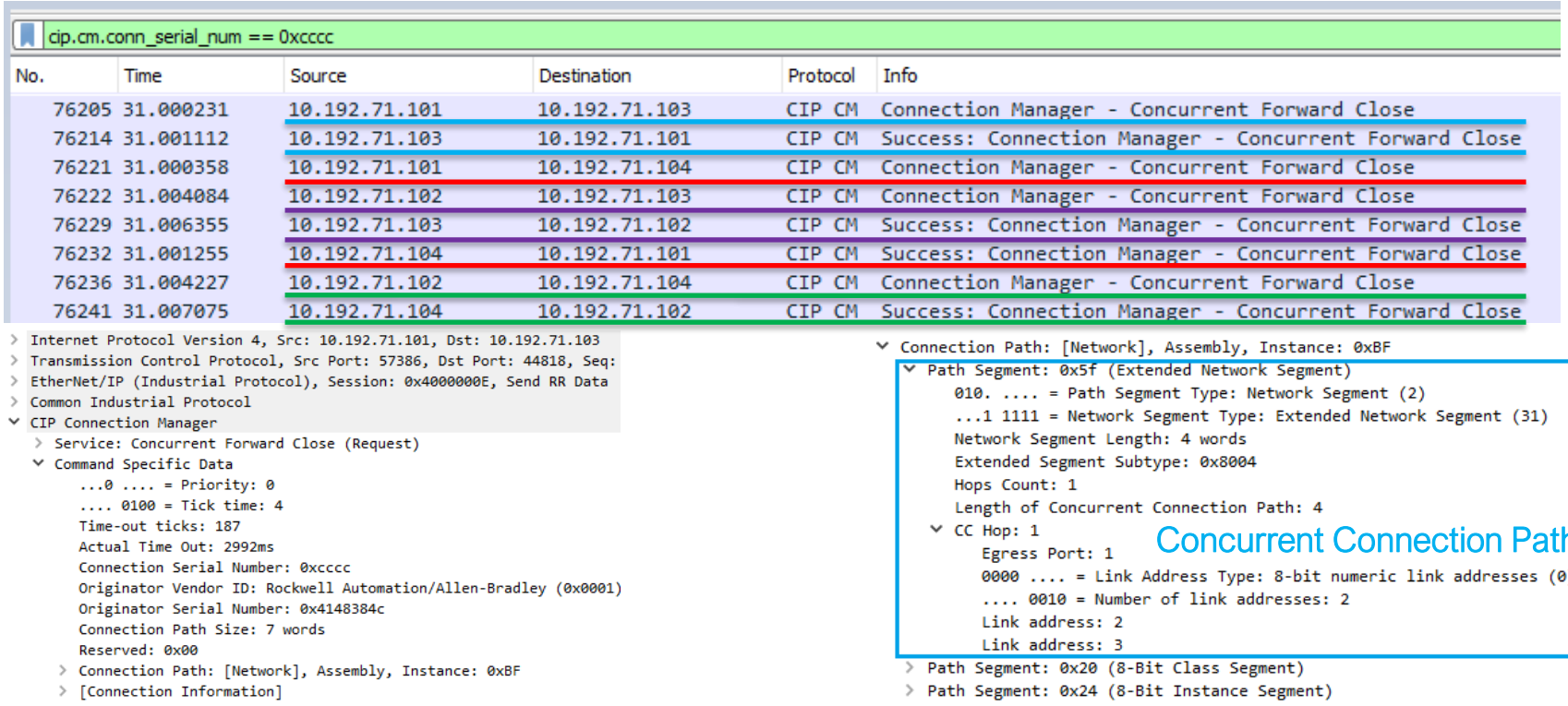
Real example – Concurrent Connection Path

```
Connection Path: [Network], Assembly, Instance: 0xBF, Connection Point: 0xC7, Connection Point: 0x0434
  Path Segment: 0x5f (Extended Network Segment)
      010. .... = Path Segment Type: Network Segment (2)
      ...1 1111 = Network Segment Type: Extended Network Segment (31)
      Network Segment Length: 4 words
      Extended Segment Subtype: 0x8004
      Hops Count: 1
      Length of Concurrent Connection Path: 4
    CC Hop: 1
        Egress Port: 1
        0000 .... = Link Address Type: 8-bit numeric link addresses (0)
        .... 0010 = Number of link addresses: 2
        Link address: 2
        Link address: 3
  Path Segment: 0x20 (8-Bit Class Segment)
  Path Segment: 0x24 (8-Bit Instance Segment)
  Path Segment: 0x2c (8-Bit Connection Point Segment)
  Path Segment: 0x2d (16-Bit Connection Point Segment)
```

Concurrent Connection Path



Real example – Concurrent_Forward_Close

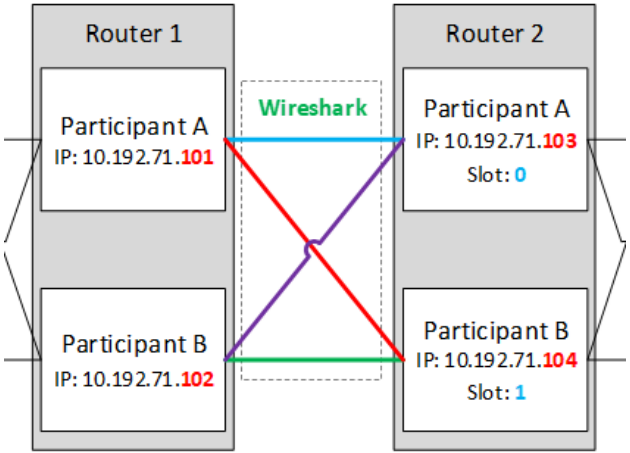


Real example – Real-Time data O->T

cip.cc.packet.sequence_count == 31900

No.	Time	Source	Destination	Protocol	Info
124	1.105903	10.192.71.101	10.192.71.104	CIP I/O	CC_SEQ=0000031900
112	1.105912	10.192.71.101	10.192.71.103	CIP I/O	CC_SEQ=0000031900
123	1.105465	10.192.71.102	10.192.71.104	CIP I/O	CC_SEQ=0000031900
111	1.105486	10.192.71.102	10.192.71.103	CIP I/O	CC_SEQ=0000031900

```
> Internet Protocol Version 4, Src: 10.192.71.101, Dst: 10.192.71.104
> User Datagram Protocol, Src Port: 2222, Dst Port: 2222
> EtherNet/IP (Industrial Protocol)
v Concurrent Connection Packet
  v Packet Type and Keep-alive: 0x0001, Packet Type: Concurrent Connection Packet Format
      .... .... ...0 0001 = Packet Type: Concurrent Connection Packet Format (1)
      .... .... ..0. .... = Keep-alive Flag: 0x0
      .... ...0 00.. .... = Keep-alive Hop Count: 0
      0000 000. .... .... = Reserved: 0x00
    Packet Length: 10
    Concurrent Connection Sequence Count: 31900
    CRC: 0xcde1f4fd
> Common Industrial Protocol, I/O
```



Concurrent Connections implementation hints

Concurrent Connections extend existing CIP connections

Concurrent Connections functionality	ODVA CIP Specification Vol 1
Concurrent Connection Path Extended Network Segment	C-1.4.3.6.2, 3-7.1
Concurrent_Forward_Open Connection Manager service	3-5.6.5, 3-7.2
Concurrent_Forward_Close Connection Manager service	3-5.6.6, 3-7.3
Concurrent Connection Packet production and consumption	3-7.4.1.1, 3-7.4.3, 3-7.4.4
Branch recovery	3-7.5
Diagnostics	3-5.3.1.10, 3-5.3.1.11, 3-5.3.1.12
Originators synchronization of Concurrent Connection parameters	3-7.2
Endpoints synchronization of produced connection data	3-7.4.2

Concurrent Connections Summary

Pros

- **Standardized end-to-end solution for redundant device communication across a system with devices from multiple vendors**
- Enable high availability systems, maximize system MTTF
- Flexible
- Eliminate switchover solution deficiencies
- Relatively easy to implement (extension of existing CIP connections)

Cons

- Higher use of network bandwidth
- Additional packet processing steps
- Require active synchronization of the redundant endpoints (vendor-specific)


2023
ODVA
Industry Conference and 22nd Annual Meeting