

Introduction to Device Level Ring

Anatoly Moldovansky, Sivaram Balasubramanian and Brian Batke
Rockwell Automation

Presented at the ODVA
2009 CIP Networks Conference & 13th Annual Meeting
February 25, 2009
Howey-in-the-Hills, Florida

Abstract

There exists a class of applications in the industrial control market where a ring/linear network based on end nodes with integrated 2-port switches is more suitable than a conventional star network. Additionally, continuous system operation is required even when a single fault occurs in the end node, its network interface or cable system. Device Level Ring (DLR) is a fault tolerant network allowing realization of control systems with such requirements.

This paper provides an introduction to DLR network¹. It describes principle of operation of this network, its features, and performance. It also describes DLR protocol and presents recommendations for implementing the DLR interface.

1. Introduction

Many end user applications call for devices to be connected to the network using a linear topology. With such a topology, end devices typically have two Ethernet ports, and are connected in sequence, one device to the next. A problem with this approach is that a failure of one node or a link between two nodes causes nodes on either side of the failure to be unreachable. By using a ring protocol implemented in the end devices, these devices may be configured in a ring topology so that a single-point failure does not prevent communication between the remainder of the functioning devices.

This paper describes the DLR protocol that provides high network availability in a ring topology. The DLR protocol is intended primarily for implementation in EtherNet/IP end devices that have two Ethernet ports and embedded switch technology. It provides for fast network fault detection and reconfiguration in order to support the most demanding control applications. For example, a ring network of 50 nodes implementing the DLR protocol has the worst case fault recovery time of less than 3ms.

Since the DLR protocol operates at Layer 2 (in the ISO OSI network model), the presence of the ring topology and the operation of the DLR protocol are transparent to

¹ This paper is based on the DLR specification published in edition 1.6 of the CIP Networks Library (Volume 2: EtherNet/IP Adaptation of CIP).

higher layer protocols, such as TCP/IP and CIP, with the exception of a DLR Object that provides a DLR configuration and diagnostic interface via CIP.

A DLR network includes at least one node configured to be a ring supervisor, and any number of normal ring nodes. It is assumed that all the ring nodes have at least two Ethernet ports and incorporate embedded switch technology. Non-DLR multi-port devices – switches or end devices – may be placed in the ring, subject to certain implementation constraints. Non-DLR devices will also impact the worst-case ring recovery time.

2. Supported Topologies

The DLR protocol supports a simple, single-ring topology; it has no concept of multiple or overlapping rings. A network installation may however use more than one DLR-based ring, so long as each of the rings is isolated such that DLR protocol messages from one ring are not present on another ring. The DLR protocol may coexist with, but does not interface with, standard network protocols such as IEEE Spanning Tree Protocols (STP, RSTP, MSTP), and also with vendor-specific redundancy protocols. That is, users may construct network topologies with DLR protocol rings connected to switches that are running Spanning Tree or other ring protocols, as shown in Figure 1.

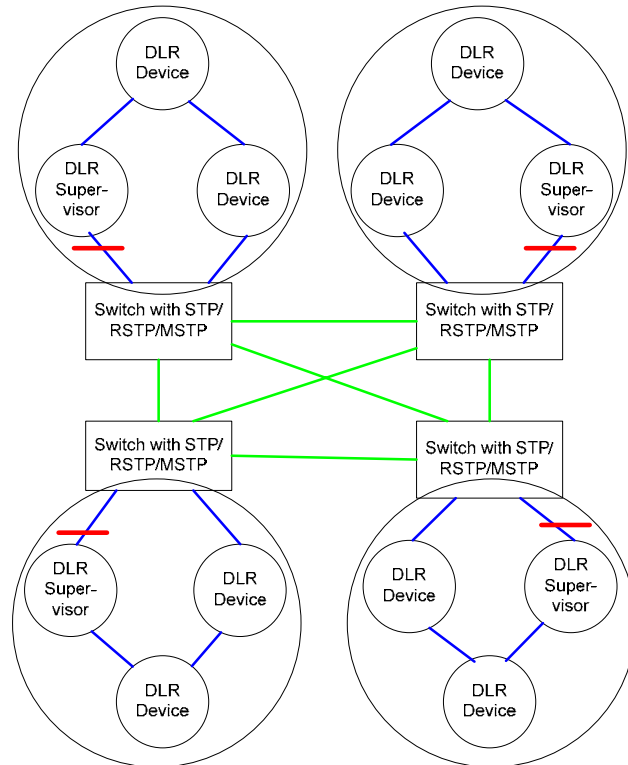


Figure 1: DLR Rings Connected to Switches

In Figure 1, each DLR ring is a separate DLR network, each with a ring supervisor. The supervisors are shown with one port in blocked mode, which is the case when there are no faults in the ring.

The switches to which the DLR rings are connected may be run STP/RSTP/MSTP to prevent loop free operation when redundant paths are present (indicated by the green lines in Figure 1). Spanning Tree Protocol messages (BPDUs) that are sent by the switch on the DLR ring ports will be blocked by the DLR Ring Supervisor (refer to Section 0 – IEEE 802.1D/802.1Q STP/RSTP/MSTP Considerations), so that the switches do not block the DLR ports.

Switch ports to which DLR devices are connected must be configured properly in order to ensure proper functioning of the network. More complicated topologies combining DLR rings and non-DLR switches running STP/RSTP/MSTP may result in DLR ports being blocked in an undesirable manner. Refer to the DLR specification for details.

3. Normal Operation

Figure 2 illustrates the normal operation of a DLR network. As it is shown there, each node has two Ethernet ports, and is assumed to have implemented an embedded switch. When a ring node receives a packet on one of its Ethernet ports, it determines whether the packet needs to be received by the ring node itself (e.g., the packet has the node's MAC address) or whether the packet should be sent out the node's other Ethernet port.

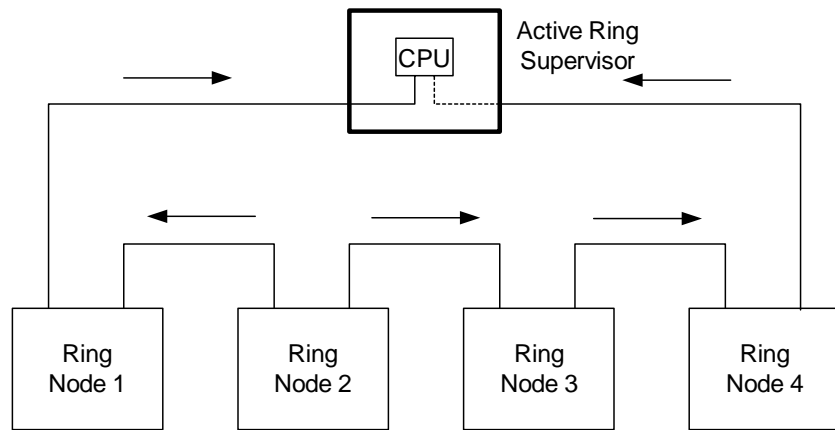


Figure 2: Normal Operation

The active ring supervisor blocks traffic on one of its ports with the exception of few special frames and does not forward traffic from one port to other. Because of this configuration a network loop is avoided and only one path exists between any two ring nodes during normal operation.

The active ring supervisor transmits a Beacon frame through both of its Ethernet ports once per beacon interval (400 microseconds by default). The active ring supervisor also sends Announce frames once per second. The Beacon and Announce frames serve several purposes:

1. The presence of Beacon and Announce frames inform ring nodes to transition from linear topology mode to ring topology mode.
2. Loss of Beacon frames at the supervisor enables detection of certain types of ring faults. (Note that normal ring nodes are also able to detect and signal ring faults).
3. The Beacon frames carry a precedence value, allowing selection of an active supervisor when multiple ring supervisors are configured.

Figure 3 illustrates the use of Beacon and Announce frames sent by the active ring supervisor:

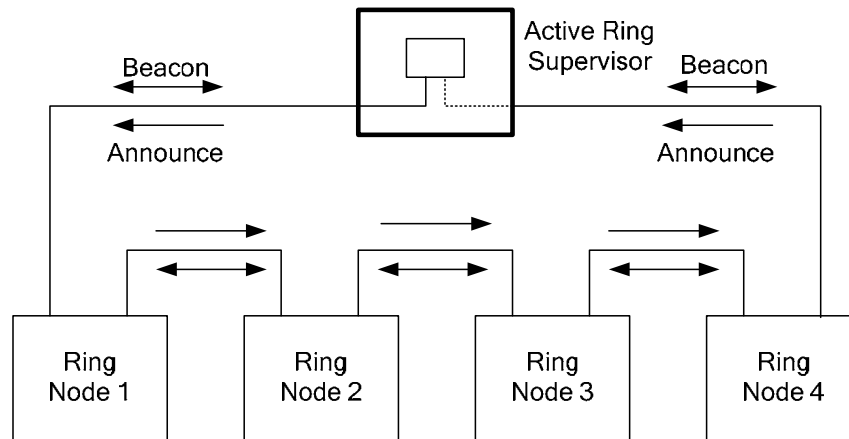


Figure 3: Beacon and Announce frames

4. Link Failures

Common Failures

The most common form of link failure includes the following cases:

- Link or other Physical layer failure recognized by a node adjacent to the failure.
- Power failure or power cycling a ring node, recognized by the adjacent node as a link failure.
- Intentional media disconnect by user to bring new nodes online or to remove existing ones.

In the above cases, the nodes adjacent to the fault send a Link_Status message to the active ring supervisor. Figure 4 shows ring nodes adjacent to a fault sending a Link_Status message to the active ring supervisor.

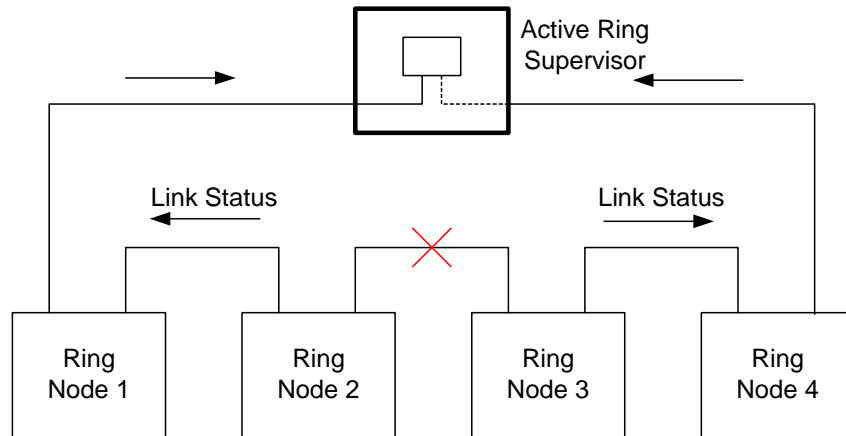


Figure 4: Link Failure

After receipt of the Link_Status message, the active ring supervisor reconfigures the network by unblocking traffic on its previously blocked port and flushing its unicast MAC table. The supervisor immediately sends Beacon and Announce frames with the ring state value indicating that the ring is now faulted.

Ring nodes also flush their unicast MAC tables upon detecting loss of the beacon in one direction, or upon receipt of Beacon or Announce frames with the ring state value indicating the ring fault state. Flushing the unicast MAC tables at both supervisor and ring nodes is necessary for network traffic to reach its intended destination after the network reconfiguration.

Figure 5 shows the network configuration after a link failure, with the active ring supervisor passing traffic through both of its ports.

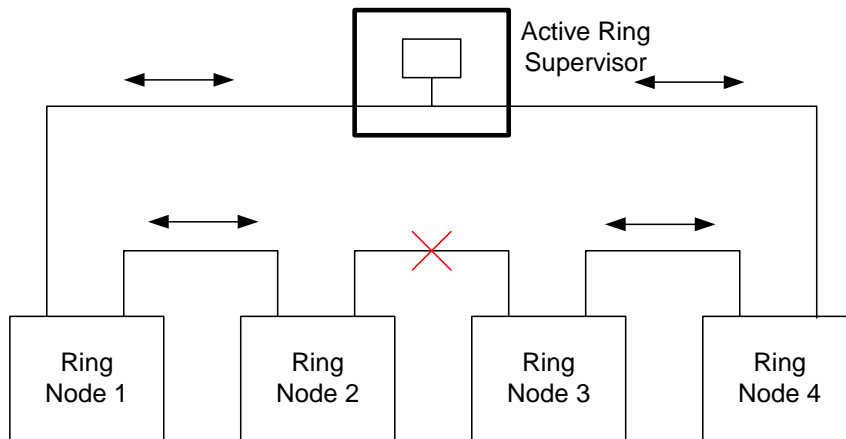


Figure 5: Network Reconfiguration after Link Failure

Uncommon Failures

In addition to the more common link failures, there is a class of uncommon failures:

- Higher level hardware/firmware component(s) on a ring node has failed leading to lost traffic, but the Physical layer is functioning normally with power supply intact.
- A chain of ring protocol unaware nodes are connected between protocol-aware nodes, and the failure has occurred somewhere in the middle of this chain.

In these cases, the active ring supervisor will detect the loss of Beacon frames first on one port, and eventually on both of its ports. The active ring supervisor will reconfigure the network as described in the “Common Failures” section. In addition, the active ring supervisor will send a Locate_Fault frame to diagnose the fault location (refer to the subsequent section on the Neighbor Check process).

It is possible for a partial network fault to occur such that traffic is lost in only one direction. The active ring supervisor detects a partial fault by monitoring the loss of Beacon frames on one port. When a partial fault is detected the active ring supervisor blocks traffic on one port and sets a status value in the DLR Object. The ring at this point will be segmented due to the partial fault, requiring user intervention.

Certain conditions such as a faulty network connector may cause the active ring supervisor to detect a series of rapid fault/restore cycles. If left to persist, such a condition could result in network instability that is difficult to diagnose. When the active ring supervisor detects the rapid fault/restore condition (5 faults in a 30 second period), it sets a status value in the DLR Object, and blocks traffic on one port. The user must explicitly clear the condition via the DLR Object.

5. Classes of DLR Implementation

There are several classes of DLR implementation, as described below. Detailed requirements for each class of implementation are further specified in subsequent sections.

Ring Supervisor

This class of devices is capable of being a ring supervisor. Such devices must implement the required ring supervisor behaviors, including the ability to send and process Beacon frames at the default beacon interval of 400 microseconds.

Ring Node, Beacon-based

This class of devices implements the DLR protocol, but without the ring supervisor capability. The device must be able to process and act on the Beacon frames sent by the ring supervisor. Beacon-based ring nodes will support beacon rates of 400 microseconds.

Ring Node, Announce-based

This class of devices implements the DLR protocol, but without the ring supervisor capability. In order to accommodate nodes that do not have the capacity to process Beacon frames, ring nodes may simply forward, but not explicitly process, Beacon frames. Such nodes must process Announce frames.

6. DLR Behavior

DLR Variables

Table 1 summarizes variables used in the DLR protocol behavior and messages. Refer to the subsequent sections on Ring Node and Ring Supervisor behavior and DLR specification for further details. The DLR Object exposes these variables (with the exception of the Node State) via object attributes.

Table 1: DLR Variables

DLR Variable	Description
Node State	Internal state of a node DLR state machine: IDLE_STATE – initial state for non-supervisors, indicating linear topology mode. FAULT_STATE – initial state for enabled ring supervisor, or when ring fault has been detected (both supervisor and ring nodes). NORMAL_STATE – normal function in ring topology mode.
Ring State	State of the ring network. Transmitted by ring supervisors in Beacon and Announce frames: RING_NORMAL_STATE – Ring is functioning, with supervisor blocking traffic on one port. RING_FAULT_STATE – Fault detected, ring supervisor is not blocking traffic (also is the initial state transmitted in the Beacon and Announce frames).
Beacon Interval	Interval at which the ring supervisor sends Beacon frames. Supervisors will support a range from 400 microseconds to 100 milliseconds. The default value will be 400 microseconds. Supervisors may support a Beacon Interval smaller than 400

	microseconds, but this is not required. The absolute minimum Beacon Interval is 100 microseconds. Beacon-based ring nodes will support beacon rates of 100 microseconds to 100 milliseconds in order to accommodate all ring supervisor implementations.
Beacon Timeout	Amount of time nodes will wait before timing out reception of Beacon frames and taking the appropriate action (depending on whether supervisor or normal ring node). Supervisors will support a range from 800 microseconds to 500 milliseconds. The default will be 1960 microseconds. Supervisors may support a Beacon Timeout of smaller than 800 microseconds but this is not required. The absolute minimum Beacon Timeout is 200 microseconds.
Supervisor Precedence	Precedence value assigned to a ring supervisor, and transmitted in Beacon frames. Used to select active ring supervisor when multiple supervisors have been configured. Default value is 0. Can be changed via the DLR Object.
DLR VLAN ID	VLAN ID used when sending DLR protocol frames. The VLAN ID is configured at the ring supervisor (via the DLR Object), and is then detected by ring nodes when they receive and process the Beacon or Announce frames from the supervisor. Default value is 0. Typically the VLAN ID does not need to be changed unless a commercial switch is being used in the ring.

Ring Supervisor

Startup

An enabled ring supervisor will start in `FAULT_STATE` and configure both ports to forward frames. The supervisor will send Beacon frames out both of its ports, with the Ring State set to `RING_FAULT_STATE`. The supervisor will also send Announce frames out both of its ports with the Ring State set to `RING_FAULT_STATE`.

Once the Beacon frames are received through both ports the supervisor will transition to `NORMAL_STATE`, flush its unicast MAC address table and reconfigure one of its ports not to forward packets, except for the following, which will be forwarded to the host for processing:

- Beacon frames with the supervisor's own MAC address (in general needed only for software implementations).
- Beacon frames from other ring supervisors.

- Link_Status/Neighbor_Status frames.
- Neighbor_Check request or response, and Sign_On: always forward received frames. For frames originated by the supervisor, only forward frames with the Source Port matching the blocked port.

Upon transition to NORMAL_STATE, the Ring State in the Beacon frames will be set to RING_NORMAL_STATE. The ring supervisor will also send an Announce frame out one port, with Ring State set to RING_NORMAL_STATE.

Multiple ring Supervisors

When multiple ring supervisors are configured, each supervisor sends Beacon frames when it comes online. The Beacon frames carry a supervisor precedence value. When a supervisor receives a Beacon frame, it checks the precedence value. If the precedence in the Beacon frame is higher than the receiving node's precedence value, the receiving node transitions to FAULT_STATE and becomes a backup supervisor. If the precedence values are the same, the node with the numerically higher MAC address becomes the active supervisor.

The backup supervisors configure their DLR parameters with the values obtained from the active supervisor's Beacon frames: Beacon Interval, Beacon Timeout, VLAN ID. The backup supervisors continue to monitor both ports for timeout of the Beacon frames (no Beacons received within the Beacon Timeout period). If the Beacon has timed out on both ports, the backup supervisor waits for an additional Beacon Timeout period (during which time other nodes transition to linear mode), then begins sending its own Beacons so that a new supervisor can be selected.

Sign on

In order to identify ring protocol participants, the active ring supervisor will send a Sign_On frame when it transitions to NORMAL_STATE.

Normal Ring Operation

When in the NORMAL_STATE, the active ring supervisor will send Beacon frames out both of its ports. It will also send an Announce frame once per second out one port

One of the active supervisor's ports will be configured not to forward frames, with the exceptions as noted in Section 0.

Ring Fault Detection

One of several possible events will cause the active ring supervisor to transition to FAULT_STATE:

- Beacon frame received from another supervisor with a higher precedence value.

- Loss of Beacon frames on either port for the period specified by the Beacon Timeout, indicating a break somewhere in the ring.
- Detection of loss of link with the neighboring node on either port.
- Link_Status frame received from a ring node, indicating a ring node has detected a fault.

In all of the cases listed above, the active ring supervisor will:

- Transition to FAULT_STATE
- Flush its unicast MAC address table
- Unblock the blocked port
- Send Beacon frame out both ports, with Ring State set to RING_FAULT_STATE
- Send Announce frame out both ports, with Ring State set to RING_FAULT_STATE

In addition, in case 2 above, the active ring supervisor will initiate the Neighbor Check process by issuing a Locate Fault frame. The active ring supervisor will also issue its own Neighbor Check frame through the port(s) on which the beacon has timed out.

When in FAULT_STATE the ring supervisor will continue to send Beacon frames, in order to detect ring restoration.

Ring Restoration

When the active ring supervisor is in FAULT_STATE, receipt of Beacon frames on both ports will cause a transition to NORMAL_STATE. The active ring supervisor will do the following:

- Flush the unicast MAC address table
- Reconfigure its ports such that traffic is not forwarded on one port (with exceptions as noted previously)
- Send Beacon frames with the Ring State set to RING_NORMAL_STATE
- Send Announce frames out one port with Ring State set to RING_NORMAL_STATE

Changing Ring Parameters

The following ring supervisor parameters may be changed via the DLR Object:

- Supervisor precedence value
- Beacon interval
- Beacon timeout
- VLAN ID
- Supervisor enabled/disabled

When any of the above parameters are changed on the active ring supervisor, the ring supervisor will cease sending Beacon frames for two beacon timeout periods then will

send Beacon frames using the new parameters. Ceasing the Beacon and Announce frames allows ring nodes to detect the new parameters when the Beacon is restored.

When parameters are changed on a backup ring supervisor, the behavior depends on the backup supervisor's new precedence value compared to the active supervisor's precedence value:

- New backup precedence value is greater than the current active ring supervisor's precedence or of equal precedence with numerically higher MAC address than active supervisor MAC address: backup will immediately begin sending Beacon frames with the new parameters.
- New backup precedence value is less than the active supervisor's precedence or of equal precedence with numerically lower MAC address than active supervisor MAC address: modification to the Beacon Interval, Beacon Timeout, and VLAN ID will be ignored.

Ring Node

Beacon VS. Announce-Based Implementations

Ring nodes (that is, non-supervisor nodes) may have different implementations depending on whether or not they are able to process the Beacon frames which by default are sent every 400 microseconds. Nodes that are able to process the Beacon frames generally have hardware assistance in implementing the DLR protocol, so that they don't burden the device's CPU with processing the Beacon frames.

Devices that would need to process the Beacon frames in the device's CPU can instead configure their embedded switch to simply pass the Beacon frames on the network without interpretation or further processing. Such devices must however process the Announce frames, which also indicate the ring state but are sent at a much slower rate.

Note that it is possible to implement a Beacon-based node without hardware assistance, provided the device's CPU has sufficient capacity to process the Beacon frames in addition to its other required functions.

It is desirable for device implementations to be Beacon-based rather than Announce-based, since better ring recovery performance results when ring nodes are able to process Beacon frames. Refer to Section 8 – Performance Analysis.

Startup – Beacon-based

A Beacon-based ring node will start up in IDLE_STATE, which presumes the network is in linear topology mode.

Upon receiving a Beacon frame through either port, the node will transition to FAULT_STATE, which presumes the ring topology mode. The ring node will flush its

unicast MAC address table and save the ring supervisor parameters from the Beacon frame:

- Supervisor MAC address
- Supervisor precedence value
- Beacon timeout
- VLAN ID

Upon receiving Beacon frames through both ports, the node will transition to NORMAL_STATE and flush its unicast MAC address table.

Startup – Announce-Based

An Announce-based ring node will start up in IDLE_STATE, which presumes the network is in linear topology mode.

Upon receiving an Announce frame through either port, the node will transition to the ring state indicated in the Announce frame. The ring node will flush its unicast MAC address table and save the ring supervisor parameters from the Announce frame:

- Supervisor MAC address
- Ring State
- VLAN ID

Fault Detection

One of several possible events will cause a ring node to transition from NORMAL_STATE:

For Beacon-based nodes:

- Receipt of a Beacon frame with the Ring State set to RING_FAULT_STATE.
- Receipt of a Beacon frame with a different MAC address and higher precedence than the current ring supervisor.
- Loss of Beacon frames on both ports for the period specified by the Beacon Timeout, which causes the node to transition to IDLE_STATE (i.e., the topology is now linear).
- Loss of Beacon on a single port for the period specified by the Beacon Timeout.

For Announce-based nodes:

- Receipt of an Announce frame with the Ring State set to RING_FAULT_STATE
- Loss of Announce frame for the Announce timeout duration, which causes the node to transition to IDLE_STATE (i.e., the topology is now linear).

In all of the cases listed above, the ring node will:

- Flush its unicast MAC address table.

- Transition to FAULT_STATE (Exception: loss of Beacon on both ports or loss of Announce causes transition to IDLE_STATE).

Ring Restoration

For ring nodes, the process for ring restoration is the same as the Startup case described above.

Sign On Process

The Sign_On frame is used to identify all ring participants. The active ring supervisor will send a Sign_On frame when it transitions to NORMAL_STATE. The active supervisor transmits a Sign_On frame once every one minute while in NORMAL_STATE, until it receives a Sign_On that it sent out previously. Upon receiving such a frame the active supervisor will cease to send further Sign_On frames until next transition into NORMAL_STATE. The collected participant list can be accessed through the DLR Object.

The Sign_On frame is a multicast message transmitted from one port of the active ring supervisor. The receiving ring participant node traps the Sign_On frame and forwards it only to the host CPU. The host CPU increments the number of nodes in list, add its own addresses to list and transmit the Sign_On frame only through the other port than the receiving port.

The Sign_On frame is transmitted from one ring participant node to the next in similar fashion and eventually reaches the active ring supervisor. The active ring supervisor can identify the Sign_On frame it sent out by confirming that the first entry is its own.

It is possible that the number of nodes in the ring large enough that all nodes' addresses do not fit in the Sign_On frame. When a node receives the Sign_On frame, if adding the node's address would exceed the maximum frame size, the node does not add its address, but saves the port through which the frame was received and sends the Sign_On frame directly to the active ring supervisor.

When the active ring supervisor receives the Sign_On frame sent to its unicast MAC address, it assumes this is due to the Sign_On frame size reaching its maximum. The active ring supervisor restarts the Sign On process by sending a new Sign_On frame directly (unicast) to the node from which it received the unicast Sign_On frame.

Upon receiving the new Sign_On frame from the active ring supervisor, the ring node adds its address to the Sign_On frame. The node then sends the Sign_On frame (multicast) through its other port than the saved port.

Neighbor Check Process

When the active ring supervisor detects the loss of Beacon, it sends a Locate_Fault frame through both ports.

Upon receipt of the Locate_Fault frame, each ring node issues a Neighbor_Check request through its port on which loss of the Beacon frame was detected (Announce-based nodes send through both ports). The supervisor also issues its own Neighbor_Check request.

When any node receives a Neighbor_Check_Request frame it responds with a Neighbor_Check_Response frame through the port on which original request was received. If the node sending the Neighbor_Check_Request does not receive a response in 100ms, it will retry the request. After three retries, if no response is received the node sends a Neighbor_Status frame to the ring supervisor.

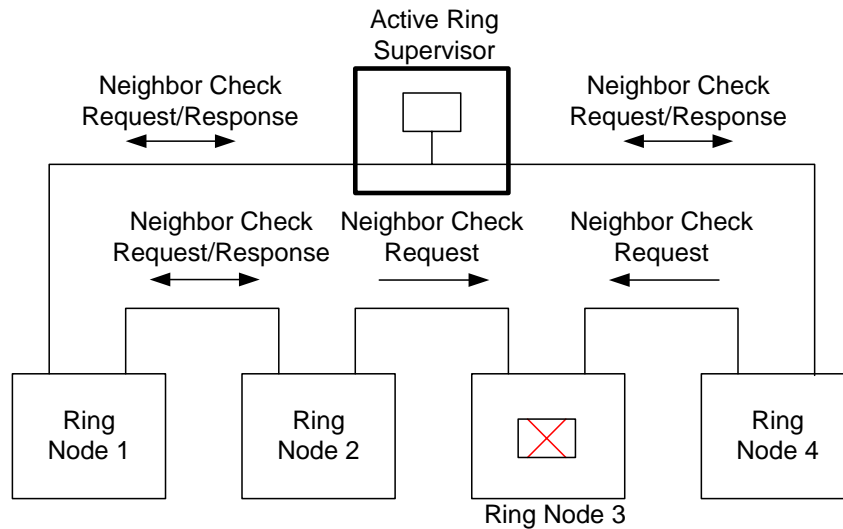


Figure 6: Neighbor Check Process

Figure 6 illustrates the Neighbor Check process. In this example, all healthy ring nodes respond, while the failed ring node 3 does not. Ring nodes 2 and 4 will each ultimately send a Neighbor_Status frame to the ring supervisor.

7. Implementation Requirements

Embedded Switch Requirements and Recommendations

The following are general requirements and recommendations for all devices that implement embedded switch technology (whether implemented via commercially-available chips, FPGA, ASIC, etc.):

- IEEE 802.3 operation:
 - Auto-negotiation, with 10/100Mbps, full/half duplex (Required)
 - Forced setting of speed/duplex (Required)
 - Recommended: Turn off flow control on ring ports
- Auto MDIX (medium dependent interface crossover), in both auto-negotiate and forced speed/duplex modes. Note: this is a PHY and transformer issue, not an embedded switch issue. (Required)
- QoS:
 - 2 queues (Required); 4 (Recommended)
 - High priority queue for DLR frames, with strict priority scheduling for the high priority queue (Required)
 - Prioritization via 802.1Q/D (Required) and DSCP (highly recommended). Usage will be consistent with the EtherNet/IP QoS scheme published in Volume 2. For IP frames the embedded switch should use the DSCP value. For non-IP frames the priority in the 802.1Q header should be used.
- Broadcast rate limiting for host CPU (Recommended). The broadcast threshold tolerated by a device is dependent on the host CPU. As a general recommendation, the broadcast rate limiting should be triggered when the broadcast traffic exceeds 1% of bandwidth.
- Filtering of incoming unicast and multicast to host CPU (Recommended, but in practice most all devices will require this).

DLR Implementation Requirements

The following implementation requirements apply to DLR nodes, whether ring supervisors or ring nodes:

- Preserve IEEE 802.1Q VLAN Id and tag priority of ring protocol frames
- Disable IP multicast filtering on ring ports or flush multicast filtering table of ring ports on ring state transitions.
- Configure multicast address for Beacon frames to be forwarded on ring ports, and to the host CPU for Beacon-based implementations.
- Configure multicast address for Announce and Locate_Fault frames to be forwarded to the host CPU and on ring ports.
- Configure multicast address for Neighbor_Check_Request /Response and Sign_On to be forwarded only to host CPU.
 - Implement a mechanism to flag the port through which such a frame was received from ring.
 - Implement a mechanism to forward such frames from host CPU on to ring only through the port it was intended to go out.

- Configure unicast MAC address of active ring supervisor so that supervisor frames forwarded on both ports
- Flush unicast MAC address tables on ring state transitions (or disable learning)
- Configure unicast MAC address of self so that it is not purged when MAC address table is flushed.
- Implement the Interface Counters and Media Counters attributes of the Ethernet Link Object to aid in network monitoring.
- Implement the QoS Object with, at a minimum, DSCP marking of EtherNet/IP traffic generated by the device.
- Recommended: configure access control list or another suitable mechanism to remove device's own frames from network when received (e.g., during ring startup/restoration)

IEEE 1588 / CIP SYNC Considerations

In order to support applications requiring time synchronization (e.g., CIP Motion or CIP Sync), multi-port devices are recommended to support the following capabilities with respect to IEEE 1588/CIP Sync:

- Implement IEEE 1588 end-to-end transparent clock.
- Devices that also implement 1588 ordinary/boundary clock functionality should perform path delay measurement using Delay_Req/Delay_Resp frames whenever the ring state or network topology mode changes.
- Devices that implement 1588 ordinary/boundary clock functionality and are connected to the ring network indirectly should perform path delay measurement using Delay_Req/Delay_Resp frames per the ODVA-specific IEEE 1588 profile signaling message.

Devices not implementing these features will suffer from poor synchronization accuracy for short periods after a network reconfiguration. Refer to the CIP Sync specification in Volume 1 of the CIP Networks Library for more information on CIP Sync.

IEEE 802.1D/802.1Q STP/RSTP/MSTP Considerations

In order for DLR to coexist with IEEE spanning tree protocols STP, RSTP and MSTP the active ring supervisor will not forward multicast frames with destination address 01:80:C2:00:00:00 (BPDU frames) from one ring port to other, irrespective of ring state. However, if the active ring supervisor has non-ring ports (e.g., in the case of a switch) it will forward said multicast frames between those non-ring ports and only between one ring port and those non-ring ports. This behavior will be implemented to ensure that any loop through the ring other than the one through active ring supervisor is detected correctly by STP/RSTP/MSTP.

8. Performance Analysis

The example performance calculations below uses following key parameters/assumptions:

Variable	Description
NumNodes	Number of nodes in DLR network, 50 for example.
BcnFrmDly	Delay due to a Beacon frame in store-and-forward switching, 7 microseconds (64 byte frame with on wire overhead) for example.
AvgEipFrmDly	Delay due to average EtherNet/IP frame, 12 microseconds (128 byte frame with on wire overhead) for example.
MaxFrmDly	Delay due to maximum size Ethernet frame, 124 microseconds (1522 byte frame with on wire overhead) for example.
MaxDlyNodePrct	Percentage of number of nodes in ring on which beacon will be delayed by maximum size Ethernet frame, 10% for example.
IntSwtchDly	Internal switching delay on every node, 5 microseconds for example.
WirePrpgtDly	Signal propagation delay on 100 meters of copper media, 1 microsecond for example.
NodeProcDly	Processing delay on nodes for responding to ring frames and events, 25 microseconds for example.
BcnIntrvl	Beacon interval, should be less than half of the fastest connection RPI, 400 microseconds for example.

In order to provide predictable performance for DLR network fault detection and reconfiguration all DLR network nodes must dedicate the highest priority queue to DLR protocol frames and must implement strict priority scheduling for highest priority queue. With such a configuration, a DLR protocol frame will encounter at most one lower priority frame delay on each node. For performance analysis, assume that all links on network operate at 100Mbps speed and in full duplex mode.

Beacon frames are 64 bytes long including frame check sequence (FCS) and have an on wire overhead of 20 bytes of which 8 bytes are for preamble and start of frame delimiter pattern and 12 bytes are for inter frame gap. Beacon frames with 20 byte on wire overhead take approximately $BcnFrmDly = 7$ microseconds on wire.

Assume that Beacon frames will be delayed on most nodes by a lower priority frame with an average size of 128 bytes including FCS. In a network with mostly EtherNet/IP traffic, on some nodes Beacon frames may not be delayed at all, in some other nodes they may be delayed by 256 byte frames and in some others it may be delayed by frames between these two extremes, for an average of 128 bytes. A 128 byte frame with 20 byte on wire overhead takes approximately $AvgEipFrmDly = 12$ microseconds on wire.

Assume that DLR nodes use store-and-forward switching architecture and that each node has an average internal switching overhead delay of $IntSwtchDly = 5$ microseconds.

Assume propagation delay for copper media of 100 meters to be $\text{WirePrpgtDly} = 1$ microsecond. The total typical delay per node for Beacon frames is therefore

$$\text{TypclDlyPerNode} = \text{BcnFrmDly} + \text{AvgEipFrmDly} + \text{IntSwtchDly} + \text{WirePrpgtDly}$$

$$\text{TypclDlyPerNode} = 7 + 12 + 5 + 1 = 25 \text{ microseconds}$$

Assume that the Beacon frames would also be delayed on $\text{MaxDlyNodePrct} = 10\%$ of nodes by maximum sized Ethernet frames of 1522 bytes each or some combination of large frames on more than 10% of nodes that is equal to 10% of nodes with maximum sized frames. Such frames may be present on network for any reason including configuration, HMI, web, etc. A 1522 byte frame with 20 byte on wire overhead takes approximately $\text{MaxFrmDly} = 124$ microseconds on wire. The total maximum delay per node for ring Beacon frames on these nodes is therefore

$$\text{MaxDlyPerNode} = \text{BcnFrmDly} + \text{MaxFrmDly} + \text{IntSwtchDly} + \text{WirePrpgtDly}$$

$$\text{MaxDlyPerNode} = 7 + 124 + 5 + 1 = 137 \text{ microseconds}$$

For a DLR network comprised of $\text{NumNodes} = 50$ nodes, total maximum round trip time for Beacon frames is therefore

$$\begin{aligned} \text{MaxRndTripTime} &= (\text{NumNodes} * (1 - \text{MaxDlyNodePrct}) * \text{TypclDlyPerNode}) \\ &+ (\text{NumNodes} * \text{MaxDlyNodePrct} * \text{MaxDlyPerNode}) \end{aligned}$$

$$\text{MaxRndTripTime} = (50 * (1 - 0.1) * 25) + (50 * 0.1 * 137) = 1810 \text{ microseconds}$$

For same network, minimum delay per node is when Beacon frame is not delayed by any other frame and therefore

$$\text{MinDlyPerNode} = \text{BcnFrmDly} + \text{IntSwtchDly} + \text{PrpgtDly}$$

$$\text{MinDlyPerNode} = 7 + 5 + 1 = 13 \text{ microseconds}$$

For a ring network comprised of $\text{NumNodes} = 50$ nodes, total minimum round trip time is therefore

$$\text{MinRndTripTime} = \text{NumNodes} * \text{MinDlyPerNode}$$

$$\text{MinRndTripTime} = 13 * 50 = 650 \text{ microseconds.}$$

In general, lower Beacon interval provides faster ring recovery performance. Beacon interval should be less than half of the fastest connection RPI in the network to prevent connection timeouts. Assume a Beacon interval of $\text{BcnIntrvl} = 400$ microseconds, which constitutes 1.75% of network bandwidth and is suitable for high performance CIP Motion connections with 1 millisecond RPI and also works for slower I/O connections.

For choosing Beacon timeout, consider a first Beacon frame transmitted at time T_{X1} facing minimum round trip time delay and arriving at active supervisor ports at time $R_{X1} = T_{X1} + \text{MinRndTripTime} = T_{X1} + 650$ microseconds. Assume that a second Beacon frame transmitted at time $T_{X2} = T_{X1} + \text{BcnIntrvl} = T_{X1} + 400$ microseconds is lost on route due to frame corruption. A third Beacon frame transmitted at time $T_{X3} = T_{X2} + \text{BcnIntrvl} = T_{X2} + 400$ microseconds facing maximum roundtrip delay will arrive at active supervisor ports at time $R_{X3} = T_{X3} + \text{MaxRndTripTime} = T_{X3} + 1810$ microseconds. The maximum arrival delay of third Beacon frame on active supervisor ports after first Beacon frame is therefore equal to $R_{X3} - R_{X1} = (T_{X3} + 1810) - (T_{X1} + 650) = (T_{X1} + 400 + 400 + 1810) - (T_{X1} + 650) = 1960$ microseconds. Hence the Beacon timeout duration should be equal to $\text{BcnTimeout} = 1960$ microseconds.

Assume end node processing delay of $\text{NodeProcDly} = 25$ microseconds for responding to DLR frames or events.

For the network described, following will be the worst case performance for DLR nodes that rely on the Beacon frame mechanism:

1. Faults that are detectable in Physical layer: This is the most common type of faults. The worst case scenario happens when the fault occurs half way across DLR network from the active ring supervisor. It will take half round trip time for `Link_Status` frames to reach from neighboring nodes of fault to active ring supervisor. It will take another half round trip time for Beacon frame with `FAULT_STATE` from the active ring supervisor to reach farthest node or for ring nodes to timeout Beacon frames whichever happens earlier. End node processing delay is involved for three times, once at fault neighbor, once at active ring supervisor and once at farthest node. The total worst case delay is therefore

$$\text{PhyscLyrFltDlyBcn} = (2 * 0.5 * \text{MaxRndTripTime}) + (3 * \text{NodeProcDly})$$

$$\text{PhyscLyrFltDlyBcn} = 1810 + (3 * 25) = 1885 \text{ microseconds}$$

2. Faults that are not detectable in Physical layer: This type of faults is relatively rare. The worst case scenario happens when the fault occurs half way across DLR network from the active ring supervisor. It will take half round trip time for last Beacon frame from near fault location to reach active ring supervisor. It will take another beacon timeout period for active ring supervisor and ring nodes to timeout Beacon frames. End node processing delay is involved once at all nodes. The total worst case delay is therefore

$$\text{NonPhyscLyrFltDlyBcn} = (0.5 * \text{MaxRndTripTime}) + \text{BcnTimeout} + \text{NodeProcDly}$$

$$\text{NonPhyscLyrFltDlyBcn} = (0.5 * 1810) + 1960 + 25 = 2890 \text{ microseconds}$$

3. Network restoration to normal mode of operation: It is equal to a total of one Beacon interval for a Beacon to be generated and one maximum round trip time for Beacon. End node processing delay is involved once at all nodes. The total worst case delay is therefore

$$\text{RingRstrDlyBcn} = \text{BcnIntrvl} + \text{MaxRndTripTime}$$

$$\text{RingRstrDlyBcn} = 400 + 1810 + 25 = 2235 \text{ microseconds}$$

For the network described, following will be the worst case performance for DLR nodes that rely on Announce frame mechanism:

1. Faults that are detectable in Physical layer: This is the most common type of faults. The worst case scenario happens when the fault occurs half way across DLR network from the active ring supervisor. It will take half round trip time for Link Status frames to reach from neighboring nodes of fault to active ring supervisor. It will take another half round trip time for Announce frame with FAULT_STATE from active ring supervisor to reach farthest node. End node processing delay is involved for three times, once at fault neighbor, once at active ring supervisor and once at farthest node. The total worst case delay is therefore

$$\text{PhysclLyrFltDlyAnnc} = (2 * 0.5 * \text{MaxRndTripTime}) + (3 * \text{NodeProcDly})$$

$$\text{PhysclLyrFltDlyAnnc} = 1810 + (3 * 25) = 1885 \text{ microseconds}$$

2. Faults that are not detectable in Physical layer: This type of faults is relatively rare. The worst case scenario happens when the fault occurs half way across DLR network from active ring supervisor. It will take half round trip time for last Beacon frame from near fault location to reach active ring supervisor. It will take another Beacon timeout period for active ring supervisor to timeout Beacon frames. It will take another half round trip time for Announce frame with FAULT_STATE from active ring supervisor to reach farthest node. End node processing delay is involved twice, once at active ring supervisor and once at end node. The total worst case delay is therefore

$$\text{NonPhysclLyrFltDlyAnnc} = (2 * 0.5 * \text{MaxRndTripTime}) + \text{BcnTimeout} + (2 * \text{NodeProcDly})$$

$$\text{NonPhysclLyrFltDlyAnnc} = 1810 + 1960 + (2 * 25) = 3820 \text{ microseconds}$$

3. Network restoration to normal mode of operation: It is equal to a total of one Beacon interval for a Beacon to be generated, one maximum round trip time for Beacon and another maximum round trip time for Announce frame to reach farthest node. End node processing delay is involved twice, once at active ring supervisor and once at end node. Note that this is the total delay for an Announce based node to flush its

unicast MAC table. The ring would have been restored earlier per calculation for Beacon based nodes. The total worst case delay is therefore

$$\text{RingRstrDlyAnnc} = \text{BcnIntrvl} + (2 * \text{MaxRndTripTime}) + (2 * \text{NodeProcDly})$$

$$\text{RingRstrDlyAnnc} = 400 + (2 * 1810) + (2 * 25) = 4070 \text{ microseconds}$$

Based on similar calculations, Table 2 below provides configuration parameters and worst case performance numbers for different DLR network node numbers. It should be noted that though these performance numbers will work for most cases, deviation from assumptions outlined above will require recalculation of numbers based on the procedure described above. For example, if any network link is set to operate at 10Mbps speed and/or half duplex mode the numbers must be adjusted (for 10Mbps, multiply by 10).

Table 2: Example DLR Network Configuration Parameters and Performance

Number of Ring Nodes	Beacon Interval (usecs)	Round Trip Time ¹ (usecs)	Beacon Timeout (usecs)	Physical Layer Faults Recovery Delay ¹ (usecs)	Non-physical Layer Faults Recovery Delay for Beacon Frame Based Nodes (usecs)	Non-physical Layer Faults Recovery Delay for Announce Frame Based Nodes (usecs)	Ring Restore Delay for Beacon frame Based Nodes (usecs)	Ring Restore Delay for Announce frame Based Nodes (usecs)
25	400	905	1380	980	1858	2335	1330	2260
50 (nominal network size)	400	1810	1960	1885	2890	3820	2235	4070
100	400	3620	3120	3695	4955	6790	4045	7690
150	400	5430	4280	5505	7020	9760	5855	11310
200	400	7240	5440	7315	9085	12730	7665	14930
250	400	9050	6600	9125	11150	15700	9475	18550

¹. Same for Beacon and Announce frames based nodes.

Important: When non-DLR nodes are present in the ring, recovery and restoration delays are as they are for Announce-based nodes, provided the non-DLR nodes follow the requirements specified in the DLR specification. If non-DLR nodes don't follow the requirements, recovery and restoration delays are unpredictable.

9. Conclusions

The DLR protocol is suitable for EtherNet/IP networks. A DLR network is tolerant to all single-point failures providing high network availability in a single-ring topology.

The worst case fault recovery time in a 50-node DLR network is less than 3ms. The low fault recovery time allows utilization of DLR networks in hard real time control systems.

10. References

1. IEEE Std 802.3 – 2005, Part 3: “Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications.”
2. IEEE Std 802.1D – 2004, “Media Access Control (MAC) Bridges.”
3. IEEE Std 802.1Q – 2003, Virtual Bridged Local Area Networks.”
4. IEEE Std 802.1s – 2002, Virtual Bridged Local Area Networks – Amendment 3: Multiple Spanning Trees.”
5. The CIP Networks Library, Volume 1, “Common Industrial Protocol (CIP),” Edition 3.5, December 2008.
6. The CIP Networks Library, Volume 2, “EtherNet/IP Adaptation of CIP,” Edition 1.6, December 2008.

The ideas, opinions, and recommendations expressed herein are intended to describe concepts of the author(s) for the possible use of CIP Networks and do not reflect the ideas, opinions, and recommendation of ODVA per se. Because CIP Networks may be applied in many diverse situations and in conjunction with products and systems from multiple vendors, the reader and those responsible for specifying CIP Networks must determine for themselves the suitability and the suitability of ideas, opinions, and recommendations expressed herein for intended use. Copyright ©2009 ODVA, Inc. All rights reserved. For permission to reproduce excerpts of this material, with appropriate attribution to the author(s), please contact ODVA on: TEL +1 734-975-8840 FAX +1 734-922-0027 EMAIL odva@odva.org WEB www.odva.org

CIP, Common Industrial Protocol, CIP Motion, CIP Safety, CIP Sync, CompoNet, CompoNet CONFORMANCE TESTED, ControlNet, ControlNet CONFORMANCE TESTED, DeviceNet, EtherNet/IP, EtherNet/IP CONFORMANCE TESTED are trademarks of ODVA, Inc. DeviceNet CONFORMANCE TESTED is a registered trademark of ODVA, Inc. All other trademarks are property of their respective owners.