

CHAPTER 8

CIP Motion

Introduction

This chapter describes the implementation of CIP Motion on EtherNet/IP and extends the design recommendations described in [Chapter 3, "CPwE Solution Design—Cell/Area Zone"](#) and [Chapter 5, "Implementing and Configuring the Cell/Area Zone."](#) Motion control systems are common within cell/area zone manufacturing applications such as packaging, pick-n-place, converting, assembly, and robotics. Motion control systems primarily control the position and velocity of servo motors. To support this, the cell/area Industrial Automation and Control System (IACS) network infrastructure must be capable of the following main tasks:

- Managing time synchronization services
- Delivering data between devices in a timely manner

As noted in earlier chapters, the cell/area zone is where the IACS end devices connect into the cell/area IACS network. Careful planning is required to achieve the optimal design and performance from both the cell/area IACS network and IACS device perspective. This extension of the Converged Plantwide Ethernet (CPwE) architectures focuses on EtherNet/IP, which is driven by the ODVA Common Industrial Protocol (CIP) [see [IACS Communication Protocols, page 1-26](#)], and in particular is tested with Rockwell Automation devices, controllers, and applications. CIP Motion on EtherNet/IP uses the CIP Application layer protocol in conjunction with CIP Sync to handle the required time synchronization services on the EtherNet/IP cell/area IACS network. See [CIP Sync for Real-Time Motion Control, page 8-5](#). Additionally, CIP Motion uses standard 100-MB switched Ethernet as well as standard Layer 2 (CoS) and Layer 3 (DSCP) quality of service (QoS) services cell/area to prioritize the motion traffic above other types of traffic for timely data delivery.

This chapter outlines the key requirements and technical considerations for real-time motion control applications using CIP Motion on EtherNet/IP within the cell/area zone. This chapter includes the following topics:

- [EtherNet/IP for Motion Control, page 8-2](#)
- [CIP Sync for Real-Time Motion Control, page 8-5](#)
- [Prioritization Services—QoS, page 8-5](#)
- [EtherNet/IP Embedded Switch Technology, page 8-10](#)
- [CIP Motion Reference Architectures, page 8-11](#)
- [CIP Motion Reference Architecture Testing, page 8-28](#)
- [Design Recommendations, page 8-37](#)
- [Detailed Test Results, page 8-40](#)

EtherNet/IP for Motion Control

The EtherNet/IP network controls many applications, including I/O-to-AC drive control, human-machine interface (HMI) communications, and controller-to-controller interlocking, and data collection and integration with IT and manufacturing execution systems (MES). With CIP Sync and CIP Motion technologies, EtherNet/IP now handles real-time control for motion control applications, adding the final element required to make it provide a complete fieldbus solution.

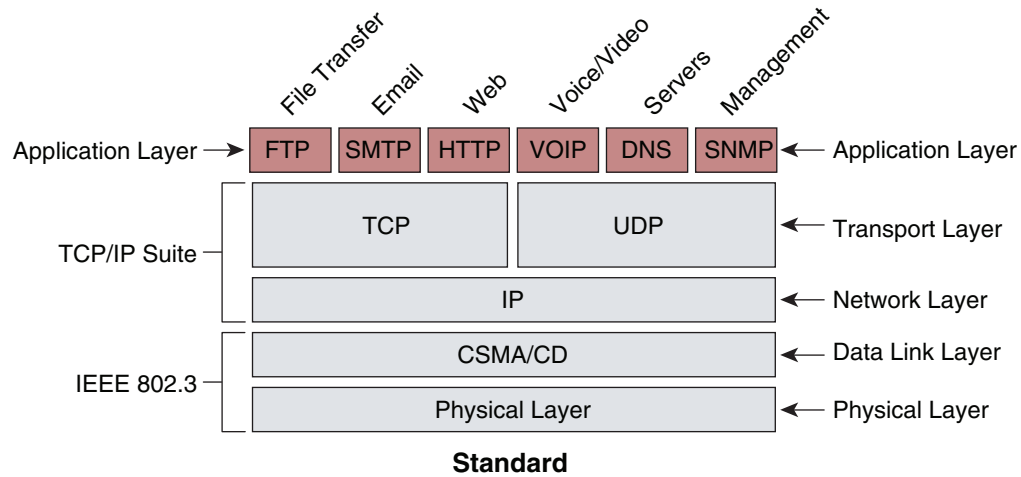
The EtherNet/IP network is not new to the industrial marketplace. Products have been shipping for more than a decade, with millions of nodes installed worldwide. EtherNet/IP network architecture is well-established. Now that real-time control for motion is available over EtherNet/IP, existing installations can absorb and incorporate this new capability with few changes to existing devices and topologies.

CIP Motion Uses Standard, Unmodified Ethernet

How can this technology be leveraged to help avoid the obsolescence of existing installations while adding real-time motion control to the network's capabilities? The key lies in its adherence to existing Ethernet standards.

In its strictest definition, the term *Ethernet* refers to the Physical and the Data Link layers of the OSI networking model; it does not, historically, refer to the Network layer, the Transport layer, or the Application layer. For this reason, many networks claim Ethernet compliance and openness despite the fact that many of the standard protocols in the other layers of the stack are not used. Although the term *Ethernet* has been applied to a very wide range of such networks, most Ethernet applications have also standardized on the Network and Transport layers of this model to communicate. Many of the software applications that exist today rely on the TCP or UDP protocols (Layer 4) as well as the IP protocol (Layer 3) to exchange data and information. These include applications such as e-mail, web pages, voice and video, and a host of other very popular applications, as shown in [Figure 8-1](#).

Figure 8-1 CIP Motion Uses Standard Network Stack Implementation

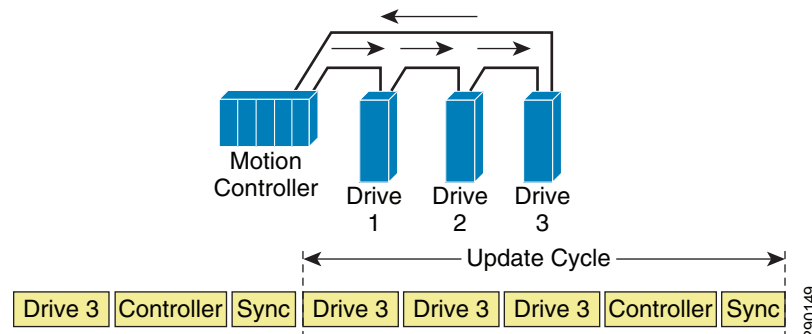


As EtherNet/IP has been developed, the standards and common protocols typically associated with Ethernet installations and applications have been maintained. In fact, CIP is an Application layer that resides on top of these layers and is portable enough to be used by the EtherNet/IP, DeviceNet, ControlNet, and CompoNet networks. Because of this, backward compatibility becomes more achievable, and current designs are more likely to avoid obsolescence. This is why real-time motion control is possible without redesigning the entire network or causing major design changes in existing EtherNet/IP installations.

Traditional Approach to Motion Control Networking

The traditional approach to handling real-time control in a motion environment is to schedule a device's time on the network. In this model, a master device establishes a marker on the network by which all other devices are synchronized (see Figure 8-2). This sync message defines the start of an update cycle. During this update cycle, the controller and drives send critical reference and feedback information. All members of the network are allocated a time slice of network time to send their data. All devices must be finished with their updates during their time slice.

Figure 8-2 Traditional Motion Approach Requires a Scheduled Network

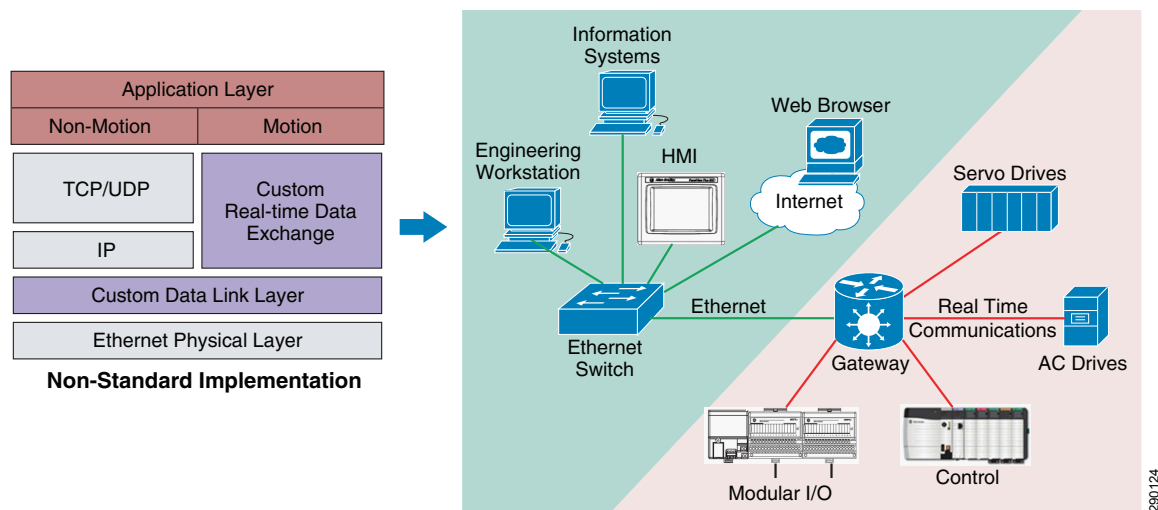


The aggregate of all the devices' time slots results in a total update cycle, which dictates the schedule that the master device uses to resend the next marker or sync message. During system configuration, as drives are added to the network, network timing is calculated, the master is given its schedule, and the timing for the system is set in place. After this schedule has been established,

no changes to data delivery can be allowed because of the predefined nature of the structure. The master sends its sync message at a deterministic interval. No tolerance exists for any jitter or deviation in communications; otherwise, the timing of the system becomes compromised.

For this reason, other real-time Ethernet solutions are forced to implement a method of scheduling Ethernet. These scheduling methods require that the Ethernet switch have special hardware and software features to support the scheduled network. These hardware and software features are dedicated to the implementation of the scheduled network and are unused by any other network applications. This means that the customer must implement a dedicated network for their motion control systems (see Figure 8-3). In addition, the sensitivity to jitter requires that the application protocol be encapsulated directly on Ethernet. These systems cannot tolerate the extra time it takes to decode IP and UDP headers. Because most industrial Ethernet implementations use TCP/IP to handle HMI and some I/O connections, this requires that the controllers and end devices maintain two network stacks, one for normal applications and one for the scheduled applications.

Figure 8-3 Use of Non-standard Stack Forces Architectural Segregation of Real-time Components



EtherNet/IP Solves Real-time Motion Control Differently

EtherNet/IP uses CIP Motion and CIP Sync to solve the problem of real-time motion control differently. CIP Sync uses the IEEE 1588 Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems, commonly referred to as the Precision Time Protocol (PTP), to synchronize devices to a very high degree of accuracy. CIP Sync incorporates the IEEE 1588 services that measure network transmission latencies and corrects for infrastructure delays. The result is the ability to synchronize clocks in distributed devices and switches to within hundreds of nanoseconds of accuracy.

When all the devices in a control system share a synchronized, common understanding of system time, real-time control can be accomplished by including time as a part of the motion information. Unlike the traditional approaches to motion control, the CIP Motion solution does not schedule the network to create determinism. Instead, CIP Motion delivers the data and the timestamp for execution as a part of the packet on the network. This allows motion devices to follow positioning path information according to a pre-determined execution plan. Because the motion controller and the drives share a common understanding of time, the motion controller can tell the drive where to go and what time to be there.

The method that CIP Motion uses to control motion is the same method used every day to attend meetings and events. In both cases, each member of the group is given information about where to go (position) and what time to be at that specific location. Because each member of the group has a watch or clock, all members arrive at the proper position at the specified time. The time at which each member receives the message about where and when to be at a location can vary. As long as the message is received early enough to allow the members to arrive on time, each member arrives in coordination with all other members. This also means that the data delivery on the Ethernet network does not need to be scheduled; only that the data must arrive just early enough to have all devices coordinate their position.

CIP Sync for Real-Time Motion Control

The previous section described how CIP Motion uses time as a necessary component of the motion information to synchronize multiple motion devices in the same system. The use of time as a part of the motion packet allows CIP Sync to coordinate time on EtherNet/IP.

CIP Sync is the name given to time synchronization services for CIP. These services allow accurate real-time synchronization of devices and controllers connected over networks that require time-stamping, sequence of events recording, distributed motion control, and other highly distributed applications that need increased control coordination.

CIP Sync is based on the IEEE 1588 standard, PTP. The standard is designed for LANs such as Ethernet, but is not limited to the Ethernet network. PTP provides a standard mechanism to synchronize clocks across a network of distributed devices.

For more information on CIP Sync technology, including such concepts as PTP, Device-level Ring (DLR), and clock types, see the Rockwell Automation publication IA-AT003, "Integrated Architecture and CIP Sync Configuration and Application Technique", at the following URL:

http://literature.rockwellautomation.com/idc/groups/literature/documents/at/ia-at003_-en-p.pdf

and Rockwell Automation publication ENET-AP005, "EtherNet/IP Embedded Switch Technology Linear and Device-level Ring Topologies Application Technique" at the following URL:

http://literature.rockwellautomation.com/idc/groups/literature/documents/ap/enet-ap005_-en-p.pdf

Prioritization Services—QoS

A traditional Ethernet network is based on the best-effort data processing mechanism, in which all the traffic gets serviced based on the first-in-first-out (FIFO) principle. However, not all network traffic is created equal, nor should users treat it equally. For example, control data (that is, CIP Sync, CIP Motion, and time-critical I/O) is more sensitive to latency and jitter than information data, such as web traffic or file transfers. To minimize application latency and jitter, control data should have priority within the cell/area zone. This prioritization is accomplished by implementing QoS, which gives preferential treatment to some network traffic at the expense of others. QoS prioritizes traffic into different service levels and provides preferential forwarding treatment to some data traffic at the expense of lower-priority traffic.

When network QoS is implemented, the network traffic is prioritized according to its relative importance and congestion-management and congestion-avoidance techniques are used to provide preferential treatment to priority traffic. Implementing QoS makes network performance more predictable and bandwidth utilization more effective.

The CPwE solution recommends a QoS implementation designed and suited for automation and control applications, including Motion and Time Synchronization. By following the recommended design and implementation guides, QoS is automatically configured to these pre-determined settings.

Customers can choose to change or modify the QoS configuration. Before modifying QoS in a particular area, use a multidisciplinary team of operations, engineering, IT, and safety professionals to establish a QoS policy. This policy should support the needs of operations, including when and where to apply QoS. Additionally, the multidisciplinary team should understand that this policy may differ from the enterprise-wide QoS policy. Enterprise-wide QoS policies commonly give priority to voice over IP (VoIP), a voice transmission protocol, which may not be as important in an individual area, and may prioritize automation and control traffic very low, thereby negatively impacting automation and control performance.

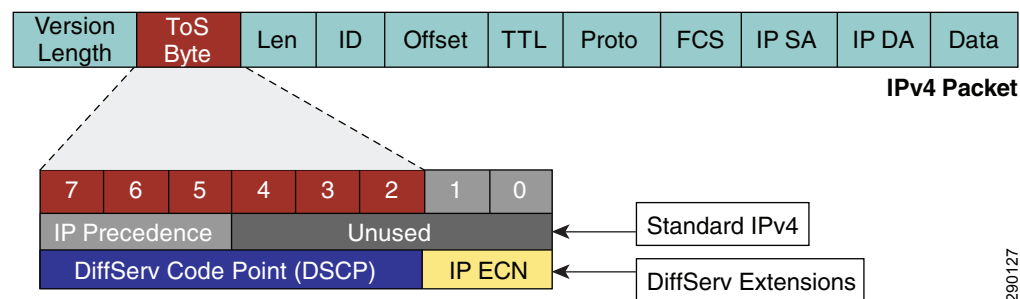
QoS Principles and Operation

This section provides an overview of the QoS concepts and techniques used for motion applications. See [Quality-of-Service \(QoS\), page 3-63](#) for more information on implementing QoS for automation and control applications.

The QoS implementation is based on the Differentiated Services (DiffServ) model, a standard from the Internet Task Force (ITF). This standard provides different treatment of traffic classes based on their characteristic behavior and tolerance to delay, loss, and jitter. The overall model is defined in RFC 2475, “An Architecture for Differentiated Services”.

DiffServ allows nodes (typically switches and routers) to service packets based on their DiffServ Code Point (DSCP) values. For CIP Motion/CIP Sync applications, the originating devices mark the DSCP values. These values are carried in the IP packet header, using the 6 upper bits of the IP type of service (ToS) field to carry the classification information, as shown in [Figure 8-4](#).

Figure 8-4 Implementation of QoS Within an IPv4 Packet



Note

Classification can also be carried in the Layer 2 frame (802.1D). However, Rockwell Automation's implementation is based on Layer 3.

As a general rule, QoS operation involves the following actions:

- **Classification**—Examining packets and determining the QoS markings (for example, DSCP and/or 802.1D priority). Many switches can also be configured to classify packets based on TCP or UDP port number, VLAN, or the physical ingress port.
- **Policing**—Per configuration, determining whether the incoming packet is within the profile or outside the profile.
- **Marking**—The incoming packet may be further marked (for example, upgraded or downgraded).
- **Queuing and scheduling**—Determining into which queue to place the packet, and servicing queues according to the configured scheduling algorithm and parameters. Switches and routers may support scheduling on a strict priority basis, round-robin basis, or other.

For more information, see the “Configuring QoS” section of the IE 3000 Software Configuration Guide, available at the following URL:

http://www.cisco.com/en/US/docs/switches/lan/cisco_ie3000/software/release/12.2_50_se/configuration/guide/swqos.html#wp1284809

In a network, the above QoS actions can be used to provide levels of service appropriate to the different needs of traffic types. Tolerance to loss, delay, and jitter are the primary factors in determining the QoS requirements for different types of traffic.

Table 8-1 shows the tolerance to loss, delay, and jitter for EtherNet/IP-related traffic.

Table 8-1 Loss, Delay, Jitter Tolerance for EtherNet/IP-related Traffic Types

Traffic Type	Traffic Characteristics	Tolerance to		
		Loss	Delay	Jitter
IEEE 1588	Fixed size messages, 44 or 64 bytes payload. Produced on a cyclic basis, once per second.	High performance applications not tolerant to loss.	PTP compensates for delays in the infrastructure.	+/-100 ns.
CIP Motion	Fixed size messages, typically 80–220 bytes. Usually produced according to a cyclic rate. High performance applications target up to 100 axes in 1 ms.	Can tolerate occasional loss of up to 3 consecutive packets. Target: 0 packet loss.	For high performance applications, less than 100 μ s.	Up to the maximum delay.
CIP I/O	Fixed size messages, typically 100 to 500 bytes. Usually produced according to a cyclic rate. Can also be produced on application change of state. Typical cyclic rate per stream: 1 to 500 ms or greater.	Application dependent. Generally can tolerate occasional loss. CIP connection typically times out if 4 consecutive packets lost. Target: 0 packet loss.	Application dependent. Tolerance proportional to the packet rate. Target: < 25% of the packet interval.	Application dependent. Generally can tolerate jitter up to the maxim tolerable delay.

Table 8-1 Loss, Delay, Jitter Tolerance for EtherNet/IP-related Traffic Types (continued)

Traffic Type	Traffic Characteristics	Tolerance to		
		Loss	Delay	Jitter
CIP Safety I/O	Fixed messages, typically on the order of 16 bytes payload. Produced according to a cyclic rate. Typical cyclic rate: 5 to 10 ms or greater.	Can tolerate occasional loss of 1 packet in a safety period (1 out of 4 transmissions).	Dependent on the packet rate; in general can tolerate delay of 5 ms.	Up to the maximum delay.
HMI Messaging	Variable size messages. Typical size: 100 to 500 bytes; likely to be larger in the future. Produced under application control. Can be at regular cyclic intervals, or based on application state or user action. Typical cyclic rate: 0.5 to 5 sec or greater.	Can tolerate packet loss so long as TCP connection remains.	Can tolerate delay as long as the TCP connection remains.	Can tolerate large degree of jitter.

Mapping CIP Traffic to DSCP and 802.1D

Based on [Table 8-1](#), different priority traffic is assigned different priority values. [Table 8-2](#) shows the default ODVA-standard priority values for CIP and IEEE 1588 traffic. The priority values can be changed via the QoS Object.

Table 8-2 DSCP and 802.1D Values

Traffic Type	CIP Priority	DSCP Enabled by Default	802.1D Priority Disabled by Default	CIP Traffic Usage
PTP event (IEEE 1588)	N/A	59 (111011)	7	PTP event messages, used by CIP Sync
PTP management (IEEE 1588)	N/A	47 (101111)	5	PTP event messages, used by CIP Sync
CIP class 0/1	Urgent (3)	55 (110111)	6	CIP Motion
	Scheduled (2)	47 (101111)	5	Safety I/O
	High (1)	43 (101011)	5	I/O
	Low (0)	31 (011111)	3	Not recommended
CIP UCMM CIP Class 3	All	27 (011011)	3	CIP messaging

QoS Support in the Infrastructure

Marking packets with DSCP or 802.1D priorities is not useful unless the network infrastructure is able to provide service based on those markings. Fortunately, most managed switches and routers support multiple queues and differentiation based on 802.1D, and many support DSCP.

Figure 8-5 and Figure 8-6 illustrate a single-queue switch and a multiple-queue switch.

Figure 8-5 Single Queue Switch

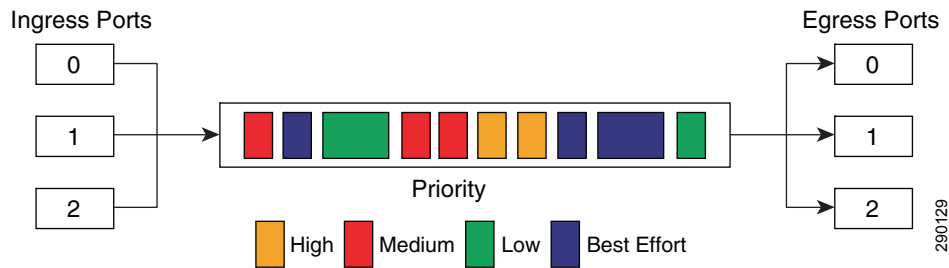
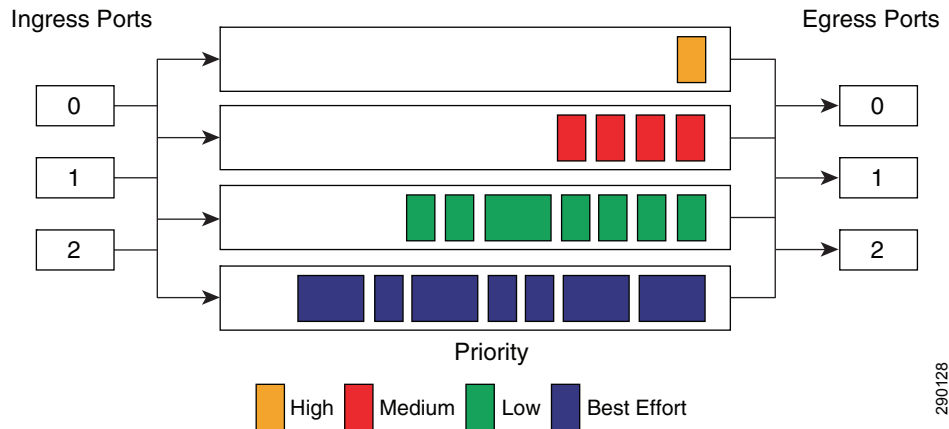


Figure 8-6 Multiple Queue Switch



In the single-queue switch, packets of all priorities are intermingled in a single queue, first-come, first-served. Higher priority packets may have to wait as lower priority packets are serviced.

In a multiple-queue switch, packets can be directed to different queues based on their priority markings. The different queues are then serviced according to a scheduling algorithm, such that higher priority packets are given precedence over lower priority packets. Often, one of the queues can be assigned strict priority where any packet in that queue is automatically serviced next.

Many switches and routers provide extensive configuration options, allowing different mappings of priorities to queues, selection of buffer space, and different scheduling algorithms (may vary by vendor).

QoS Support in the Rockwell Automation Embedded Switch Technology (DLR and Linear Topologies)

Rockwell Automation has developed a three-port switch for integration into many of its EtherNet/IP-based products. This switch has two external ports for daisy chaining and one port integrated within the product. The switch offers IEEE 1588 transparent clock functionality for re-phasing of the clocks as well as QoS functionality. It also supports the Beacon protocol, used to manage the traffic in a closed-loop ring.

The embedded switch technology enforces QoS based on IP DSCP. The embedded switch implements four prioritized transmit queues per port, as follows:

- Frames received with DSCP 59 are queued in highest priority transmit queue 1.
- Frames received with DSCP 55 are queued in second highest priority transmit queue 2.
- Frames received with DSCP 47 and 43 are queued in third highest priority transmit queue 3.
- Frames received with other DSCP values are queued in lowest priority transmit queue 4.

In addition, ring protocol frames are queued in highest priority queue 1. When a port is ready to transmit the next frame, the highest priority frame is chosen from the current set of queued frames for transmission based on strict priority ordering. Within a given priority queue, frames are transmitted in FIFO order. (See [Table 8-3](#).)

Table 8-3 Four Prioritized Transmit Queues

	Class of Service	DSCP	Notes
Highest priority	7	59	DLR/BRP, PTP Event (IEEE 1588)
High priority		55	CIP Motion
Low priority		43, 47	I/O, Safety I/O, PTP Management (IEEE 1588)
Lowest priority	1,2,3,4,5,6	0–42,44–46,48–54,56–58,60–63	Best effort



Note

Implementation of QoS in the Stratix 8000 switches differs from QoS implementation for embedded switches. For more information on QoS implementation, see [Chapter 3, "CPwE Solution Design—Cell/Area Zone."](#)

The DSCP values are aligned with the ODVA EtherNet/IP QoS object specification. The originator of a frame, including all dual-port devices, is expected to put the correct DSCP value in the frame. Legacy single-port products may not put the correct DSCP value in the frame.

EtherNet/IP Embedded Switch Technology

EtherNet/IP embedded switch technology incorporates a three-port switch (typically two external ports, and one internal port) into Rockwell Automation end devices, instead of directly to an external switch. This feature has also been incorporated into Rockwell Automation devices that support both CIP Sync and CIP Motion. The embedded switch technology is included in the 1756-EN2TR and 1756-EN3TR EtherNet/IP communication modules and the Kinetix 6500 servo drives. Each of the Kinetix 6500 control modules contain dual Ethernet ports. Examples of additional devices with embedded switch technology include Point I/O and ArmorBlock I/O products.

Products that do not support embedded switch technology can still be integrated into a linear or ring topology with a network tap (catalog numbers 1783-ETAP, 1783-ETAP1F, 1783-ETAP2F). The taps contain a single device port and two network ports to connect into a linear or ring network topology. For more information on embedded switch technology, see the Rockwell Automation publication, ENET-AP005, “EtherNet/IP Embedded Switch Technology Application Guide”, available at the following URL:

http://literature.rockwellautomation.com/idc/groups/literature/documents/ap/enet-ap005_-en-p.pdf

Support for the following EtherNet/IP embedded switch technology features are critical to both CIP Sync and CIP Motion:

- IEEE 1588 transparent clock to ensure proper time synchronization
- Beacon protocol for ring configurations
- QoS
- Internet Group Management Protocol (IGMP) for management of network traffic to ensure that critical data is delivered in a timely manner

CIP Motion Reference Architectures

The following topologies were set up and validated:

- Linear
- Ring (DLR)
- Star

Linear Topologies

This section discusses types of linear topologies.

Advantages of a linear network include the following:

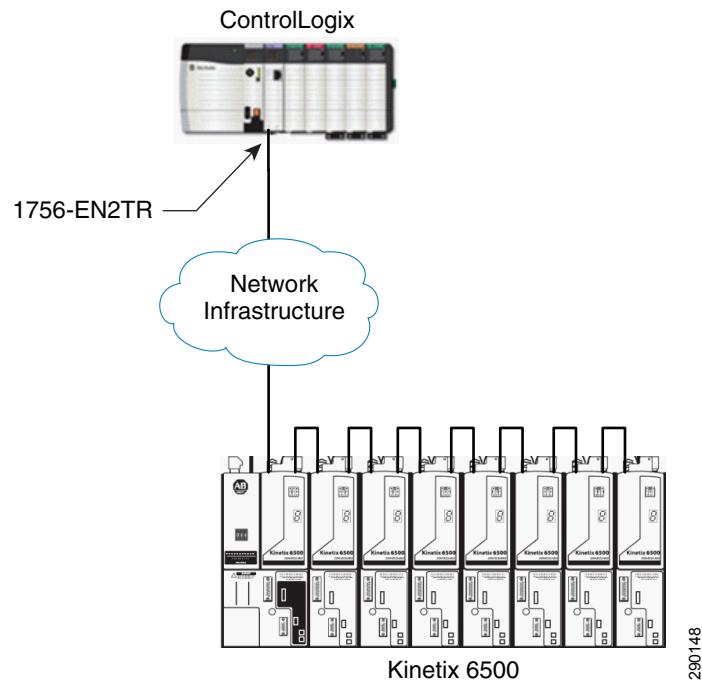
- Simplifies installation by eliminating long cable runs back to a central switch
- Extends the network over a longer distance because individual cable segments can be up to 100 m
- Supports up to 50 mixed devices per line

The primary disadvantage of a linear topology is that a lost connection or link failure disconnects all downstream devices as well. To counter this disadvantage, a ring topology can be employed.

Basic Linear Topologies

The most basic network topology for a CIP Motion network is a linear topology with a point-to-point connection between the 1756-EN2T, 1756-EN2TR, or 1756-EN3TR modules and the first Kinetix 6500 drive, as shown in [Figure 8-7](#).

Figure 8-7 Basic Linear Topology



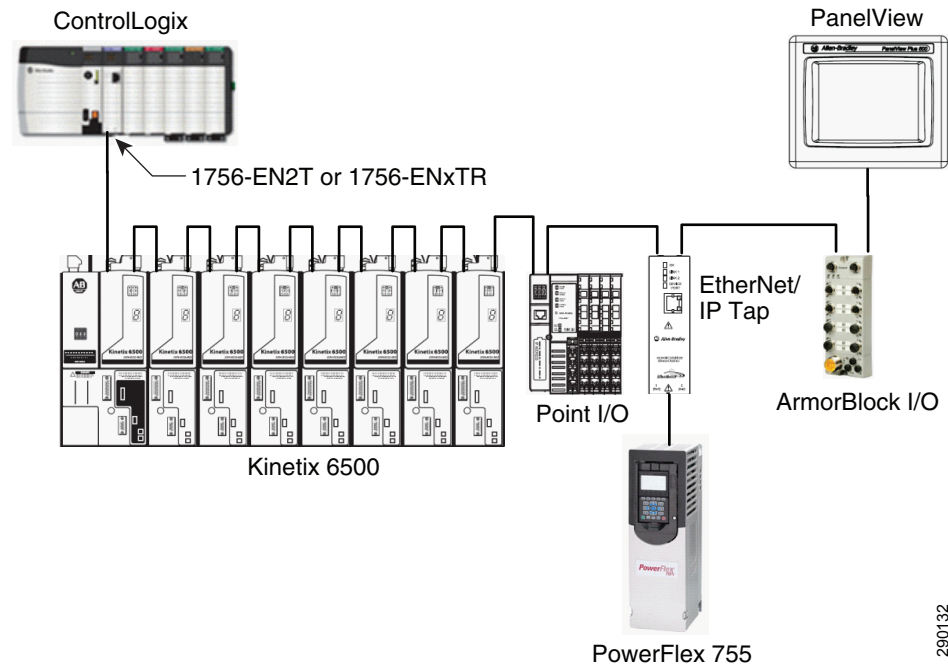
Notice that the topology shown in [Figure 8-7](#) does not require the use of an external Ethernet switch. EtherNet/IP embedded switch technology eliminates the requirement for external Ethernet switch hardware. Each of the Kinetix 6500 drives contains a dual-port switch that lets the drives be daisy-chained. Because the embedded switch technology employs a transparent clock and supports QoS and IGMP, proper time synchronization is maintained between the master and slave clocks. In addition, critical position command and drive feedback data is transmitted in a timely manner.

**Note**

The 1756-EN2T and 1756-EN2TR modules are limited to eight configured position servo drives, while the 1756-EN3TR module is limited to 128 configured position servo drives.

Linear topology can include many types of devices on the same network, as shown in [Figure 8-8](#).

Figure 8-8 Linear Topology with Additional Devices

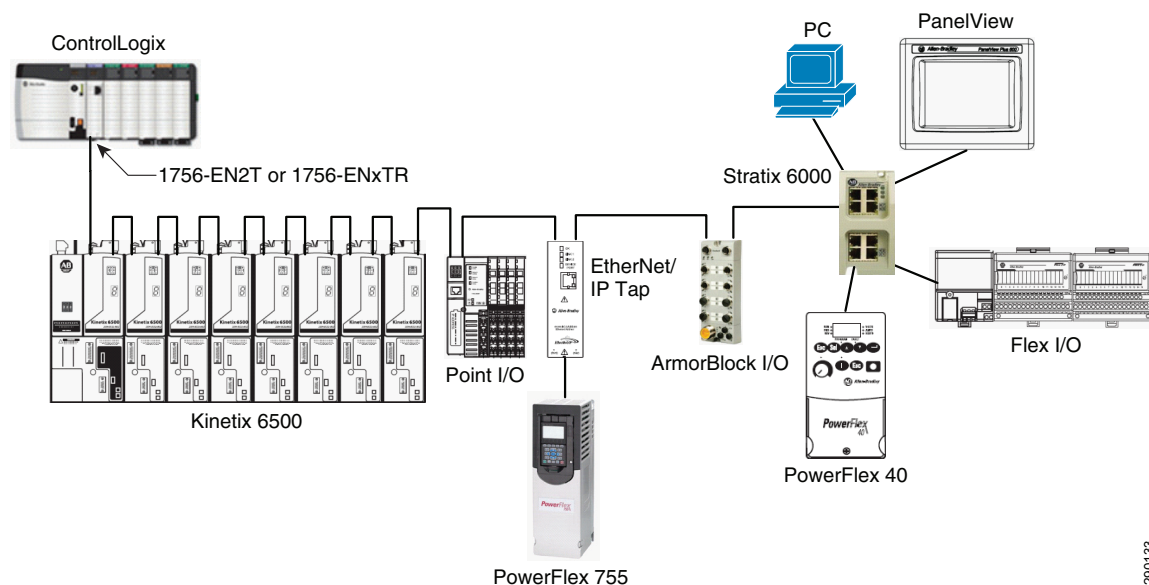


Devices with embedded switch technology such as POINT I/O or ArmorBlock I/O can be added to extend the topology. EtherNet/IP taps can be used to incorporate devices that do not contain embedded switch technology, such as the PowerFlex 755 drive. Devices that do not have embedded switch technology can also be added as the last device in the line, as illustrated by the PanelView terminal in [Figure 8-8](#).

Linear/Star Topology

Network switches can also be added to the end of the line, creating a linear/star topology, as shown in [Figure 8-9](#).

Figure 8-9 Linear/Star Topology—Linear Segment with External Switch (Star)



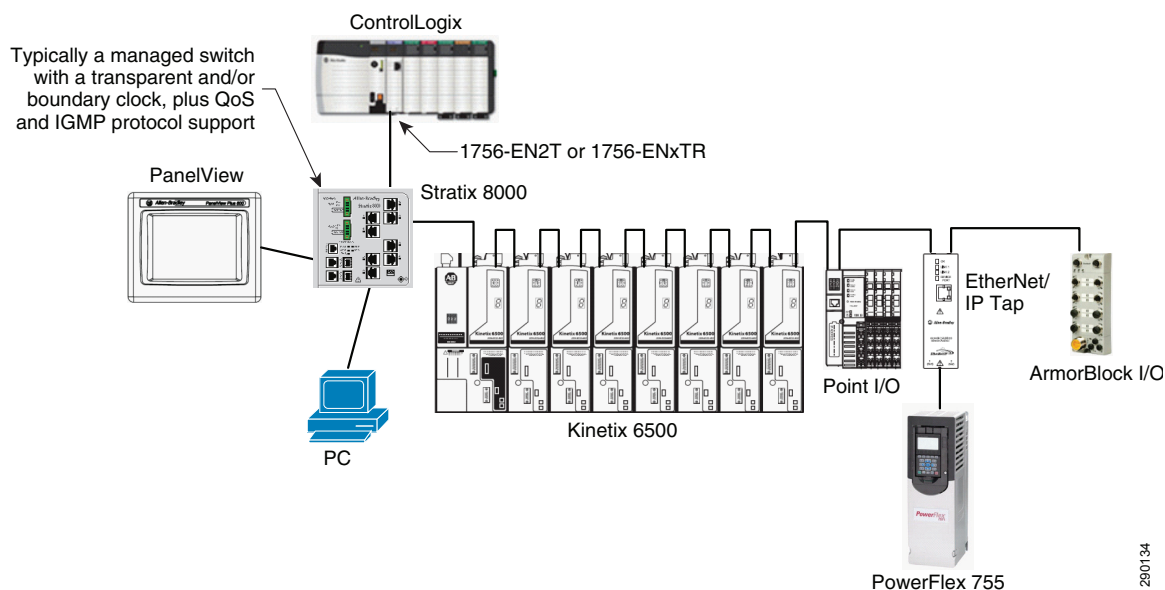
290133

Devices that do not have embedded switch technology can be connected in a star topology from the switch, as illustrated by the Stratix 6000 Ethernet switch in [Figure 8-9](#).

Star/Linear Topology

A linear segment of Kinetix 6500 drives and other devices can also be connected as a branch off of a central switch, creating a star/linear topology between the ControlLogix chassis and the first Kinetix 6500 drive, as shown in [Figure 8-10](#).

Figure 8-10 Star/Linear Topology—Linear Segment Connected to External Switch (Star)



290134

The Stratix 8000 Ethernet switch is shown because a managed switch with a transparent or boundary clock, plus QoS and IGMP protocol support, is typically required in this topology. If an unmanaged switch without these advanced features is inserted in place of the Stratix 8000 switch, time synchronization may not be maintained between master-slaves.



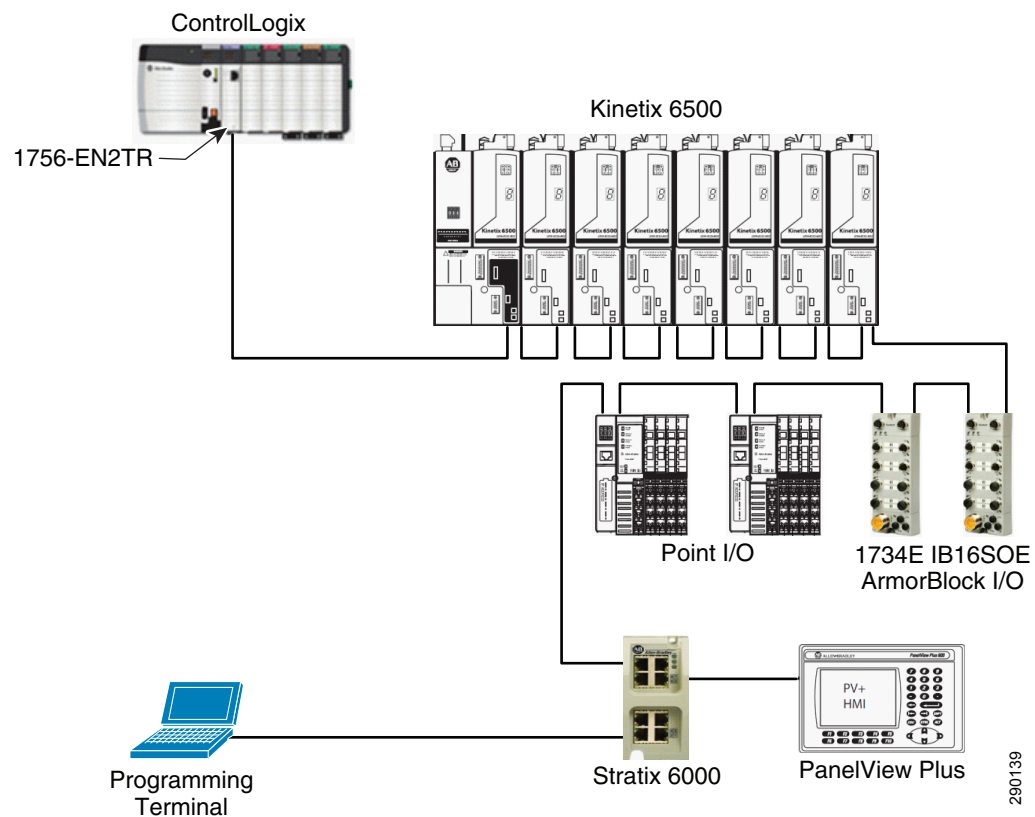
Note

The Stratix 8000 switch does not support the Beacon ring protocol; therefore, it is not recommended for use within a DLR topology. Extra care should be taken to ensure that time synchronization is not impacted. For example, an EtherNet/IP tap can be used in conjunction with an unmanaged switch to achieve a similar topology to the topology shown above, while maintaining proper time synchronization and ensuring critical data is delivered in a timely manner.

Linear Topology Reference Architectures Under Test

In this test, the time-critical components and the non-time-critical components are connected in a linear segment, as shown in [Figure 8-11](#).

Figure 8-11 Linear Reference Architecture with Switch



All the non-time-critical components, including the programming terminal and HMI (such as the PanelView Plus) are connected using the Stratix 6000 switch. The Stratix 6000 switch is connected to the end of the linear segment.

The CIP Sync packets, used for IEEE 1588 time synchronization, are exchanged between the grandmaster clock and all CIP Sync slave devices (for example, the Kinetix 6500, CIP Sync I/O) once every second. These time-critical components are equipped with transparent clocks to handle the re-phasing of time as the time sync messages pass through each device.

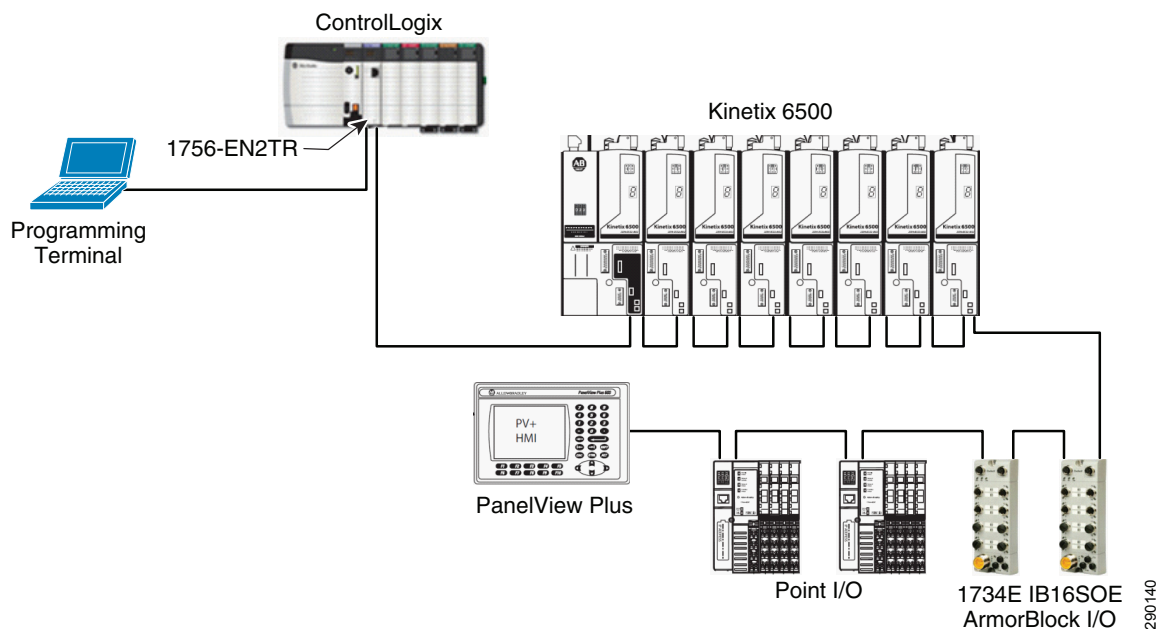
The Stratix 6000 switch, on the other hand, does not have any 1588 time-synchronization capabilities (transparent or boundary clock capabilities). This means that this switch is not capable of re-phasing time or regulating against the grandmaster in any way. Any time-synchronized packet passing through the Stratix 6000 switch experiences delays passing; and these delays vary from instance to instance, depending on such factors as traffic loading.

If the motion devices were connected to the downstream side of the Stratix 6000 switch, these delays would be directly manifested as time variations in the CIP Motion drives (depending on the delay in the switch). Depending on the application, these delays could potentially cause unacceptable disturbances in the motion system.

To avoid this effect in the motion system, the Stratix 6000 switch is connected after the Kinetix 6500 drives and I/O modules (at the end of the line). All other non-time-critical Ethernet devices are connected to the switch.

A second reference architecture for a linear topology is shown in [Figure 8-12](#). This architecture has no switch. The 1756-EN2TR or 1756-EN3TR modules in the ControlLogix chassis have two ports. These ports can be used as regular switches and can be connected to any Ethernet device for small, standalone machine architectures.

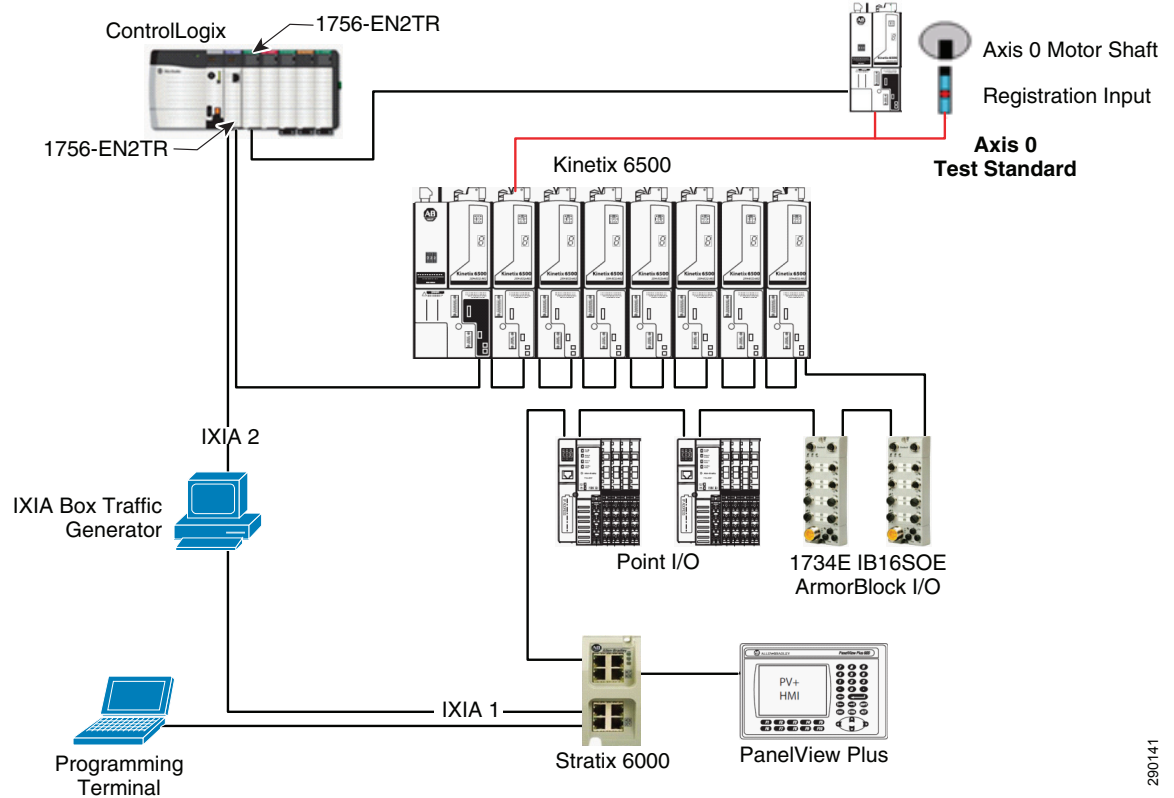
Figure 8-12 Linear Reference Architecture Without Switch



In small, standalone architectures that do not require a switch, a programming terminal or any other Ethernet device can be directly connected to a port on the 1756-EN2TR or 1756-EN3TR modules.

To test these architectures, the configuration shown in [Figure 8-13](#) was used.

Figure 8-13 Linear Test Architecture



A reference axis (indicated as Axis 0) is used as the reference for all measurements. All measurements, as discussed in [Test Criteria, page 8-29](#), are measured for this architecture.

The Ixia box is a network traffic generator device that is used to test this architecture. It generates both Class 1 and Class 3 traffic. It can also generate both multicast and unicast traffic on the network. The configuration for the Ixia box is shown in [Ixia Network Traffic Generator Configuration, page 8-33](#).

The system configuration parameters for testing this architecture are shown in [Table 8-4](#).

Table 8-4 System Configuration Parameters

Controller coarse update rate (ms)	4
Number of CIP Motion axes	8
Number of rack optimized IO	2
Number of direct IO	2
Rack optimized IO RPI (ms)	5
Direct IO RPI (ms)	1
HMI PanelView Plus	1
1783-ETAP	0

290141

DLR Topology

This section discusses DLR topology.

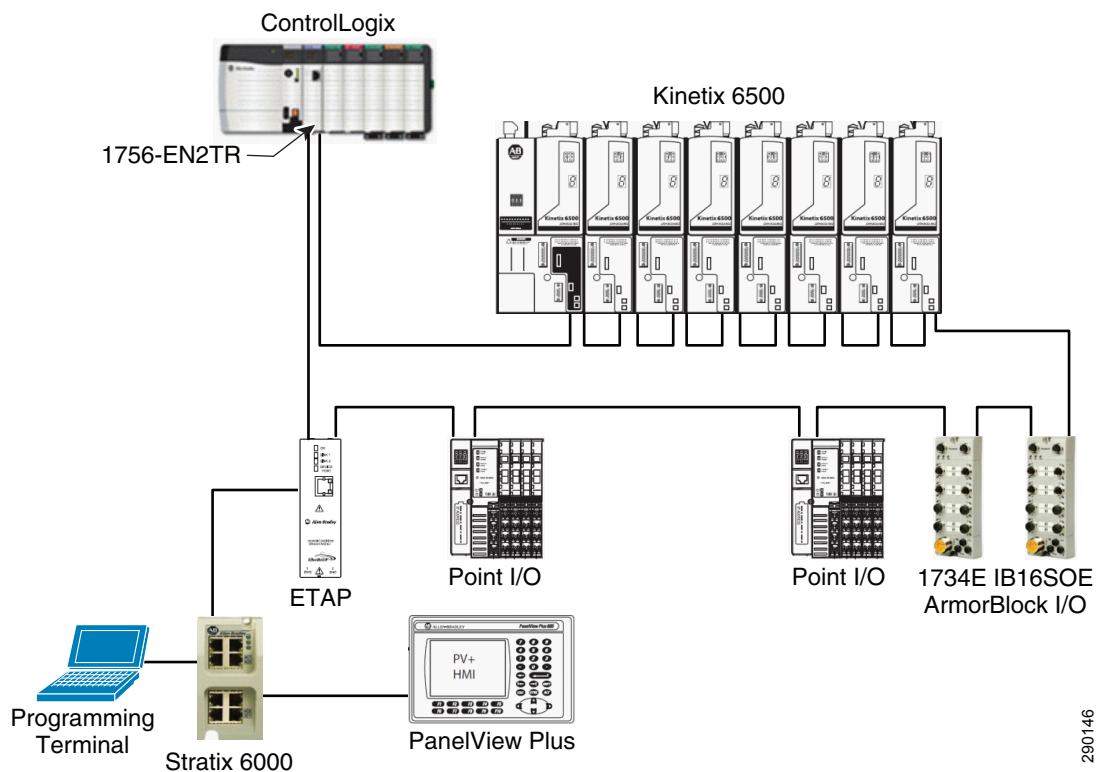
Advantages of a ring network include the following:

- Simple installation
- Resilience to a single point of failure (cable break or device failure)
- Fast recovery time from a single point of failure

The primary disadvantage of a ring topology is additional setup (for example, the active ring supervisor) over a linear or star network topology.

DLR topology is implemented similarly to a linear topology. The primary difference between the two topologies is that, with a ring topology, an extra connection is made from the last device on the line to the first, closing the loop or ring. (See [Figure 8-14](#)).

Figure 8-14 DLR Topology

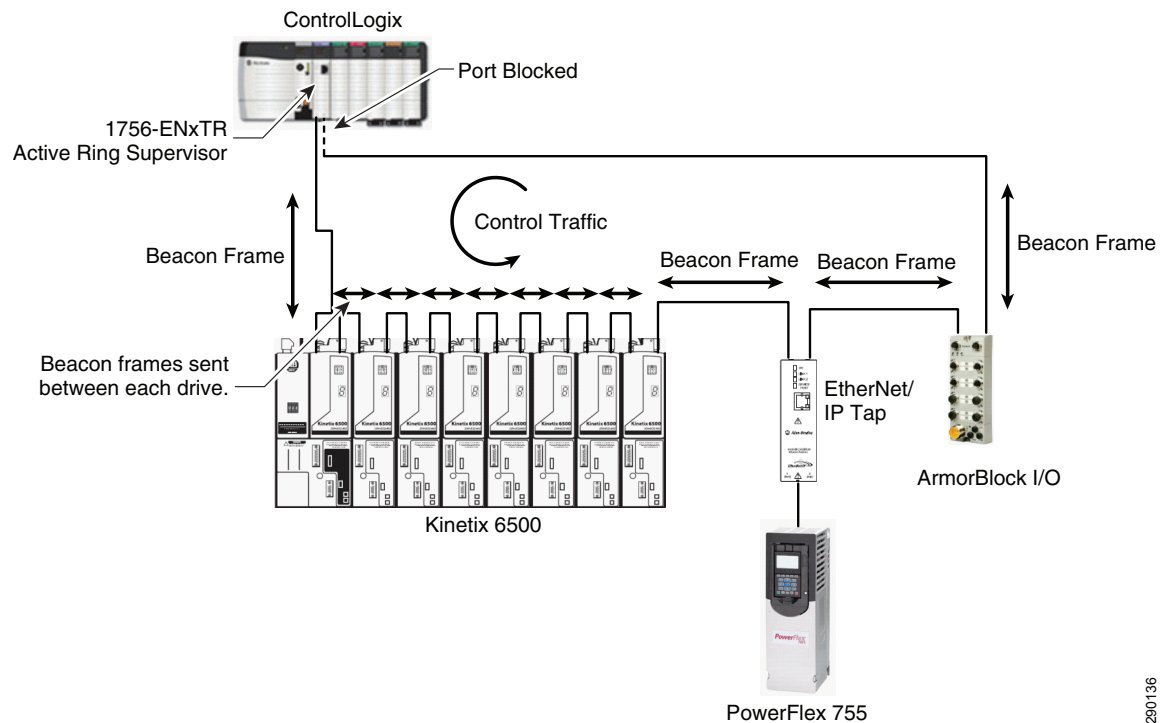


A DLR or ring topology has a distinct advantage over a linear topology. Because of the closed loop nature of a DLR, it is resilient to a single point of failure on the network. That is, if a link is broken in the DLR, the ring can recover from a single fault and maintain communications. This failure point can occur anywhere on the ring, even between daisy-chained (Kinetix 6500) drives, and the DLR is still able to recover fast enough to avoid application disruption.

One of the nodes on a DLR is considered to be the active ring supervisor; all of the other nodes can be designated as a backup supervisor or a ring node. The backup supervisor becomes the active supervisor in the event the active supervisor is interrupted or lost. The active ring supervisor is charged with verifying the integrity of the ring, and reconfigures the ring to recover from a single fault condition.

The active ring supervisor uses the Beacon protocol frames to monitor the network to determine whether the ring is intact and operating normally, or the ring is broken and faulted. During normal operation, the active ring supervisor blocks one of its ports and directs traffic in a single direction, as shown in Figure 8-15.

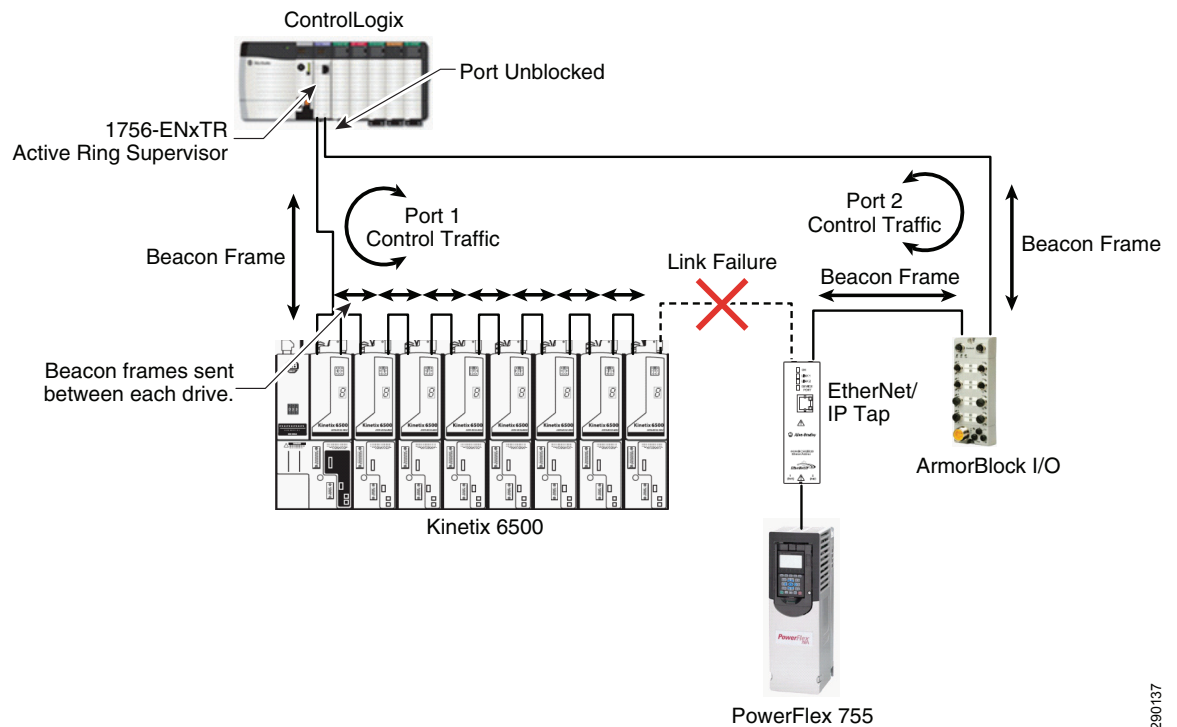
Figure 8-15 DLR Normal Operation



In the event of a network fault, based on information provided by the Beacon frames, the active ring supervisor detects the link failure and reconfigures the network to maintain communications, as shown in Figure 8-16.

290136

Figure 8-16 DLR Recovery After Link Failure

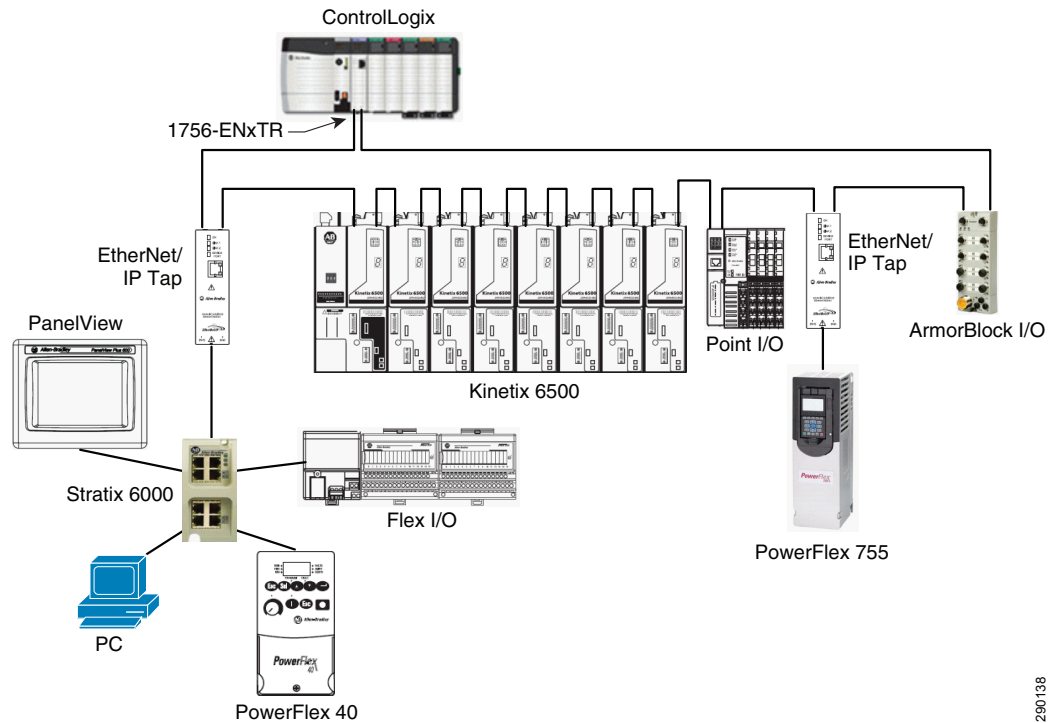


Following detection of the failure, the active ring supervisor begins passing network traffic through both of its ports by unblocking the previously blocked port. Recovery time for a DLR network is typically less than 3 ms for a 50-node network. Note that a DLR can recover only from a single point of failure.

Mixed Star/Ring Topology

Network switches can also be connected into a DLR via an EtherNet/IP tap, creating a star/ring topology, as shown in [Figure 8-17](#).

Figure 8-17 Star/Ring Topology—External Switch Connected into Ring via EtherNet/IP Tap



Devices that do not have embedded switch technology can be connected in a star topology off the switch. The DLR is able to retain all its inherent benefits, while allowing communication to occur between devices in the ring and devices connected in the star outside the ring.

DLR Topology Reference Architectures Under Test

The time-critical devices (for example, CIP Motion components, CIP Sync components, and I/O devices) are separated from the non-time-critical devices (for example, the PanelView programming terminal) in this architecture, as shown in [Figure 8-18](#).

The diagram illustrates the architecture of a Kinetix 6500 motion control system. At the top left is the **ControlLogix** 1756-EN2TR processor. To its right is a rack of eight **Kinetix 6500** modules, including a power supply and seven drives. Below the processor is an **ETAP** (Ethernet Terminal Access Point) module. A **Programming Terminal** (laptop) is connected to the ETAP. Below the ETAP is a **Stratix 6000** network switch. To the right of the ETAP is a **Point I/O** module. Below the Point I/O is a **PanelView Plus** HMI. To the right of the Point I/O is another **Point I/O** module. Below this second Point I/O are two **1734E IB16SOE ArmorBlock I/O** modules. All components are interconnected via a network backbone.

In the architecture shown in [Figure 8-18](#), the grandmaster clock is the master clock in the system. All CIP Sync-enabled devices, such as Kinetix 6500 drives and 1732E-IB16SOE Armor Block I/O modules, synchronize their clocks to the grandmaster clock. The 1756-L6x/L7x controller and 1756-ENxT modules in the ControlLogix chassis, as shown in [Figure 8-18](#), can be the grandmaster. By default, the 1756-EN2T or 1756-EN3T modules are the grandmaster when power is applied to the system.

- Eight Kinetix 6500 drives
- Two CIP Sync-enabled 1734E-IB16SOE ArmorBlock I/O modules
- Two POINT I/O adapters
- Four 1783-ETAP modules

The 1783-ETAP modules connect to all the devices that have a single port, such as a programming terminal, Ixia Box traffic generator, and Panel View Plus terminal.

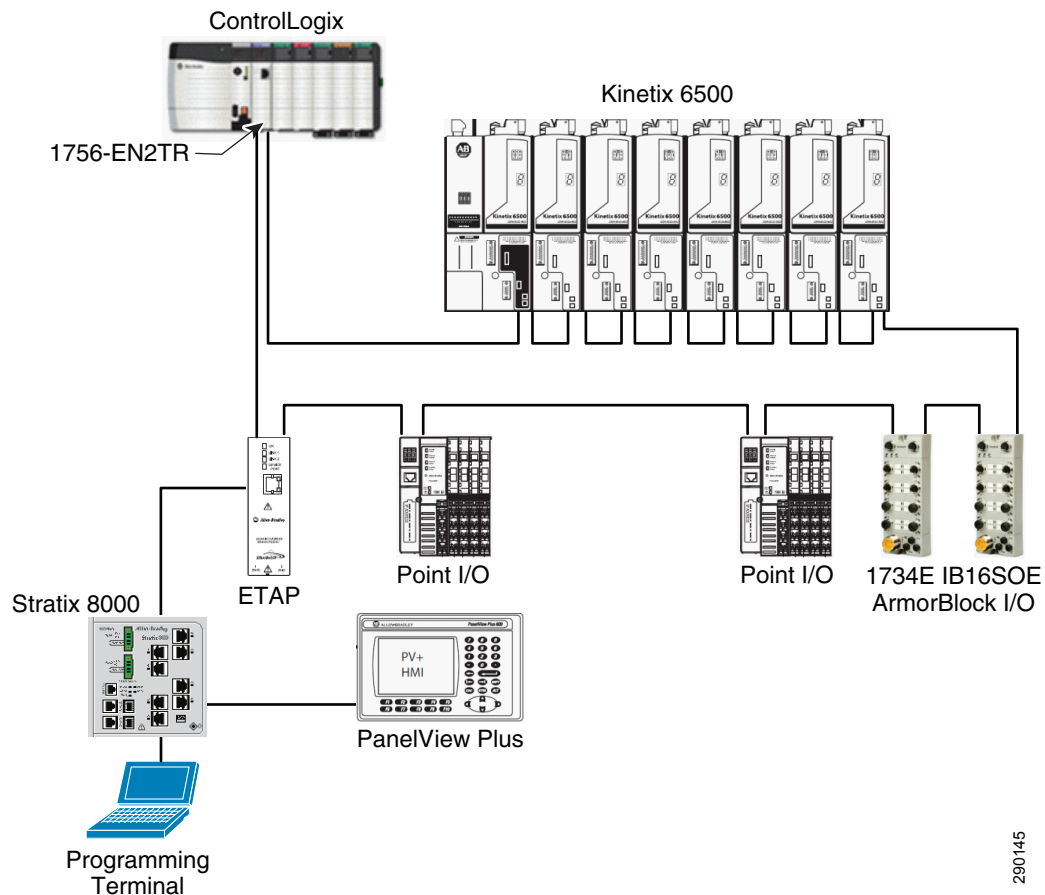
A Stratix 6000 or Stratix 8000 switch can be connected to the 1783-ETAP module, as shown in [Figure 8-19](#) and [Figure 8-20](#). The switches let you plug in many more Ethernet devices on the network without needing additional 1783-ETAP modules.



Note

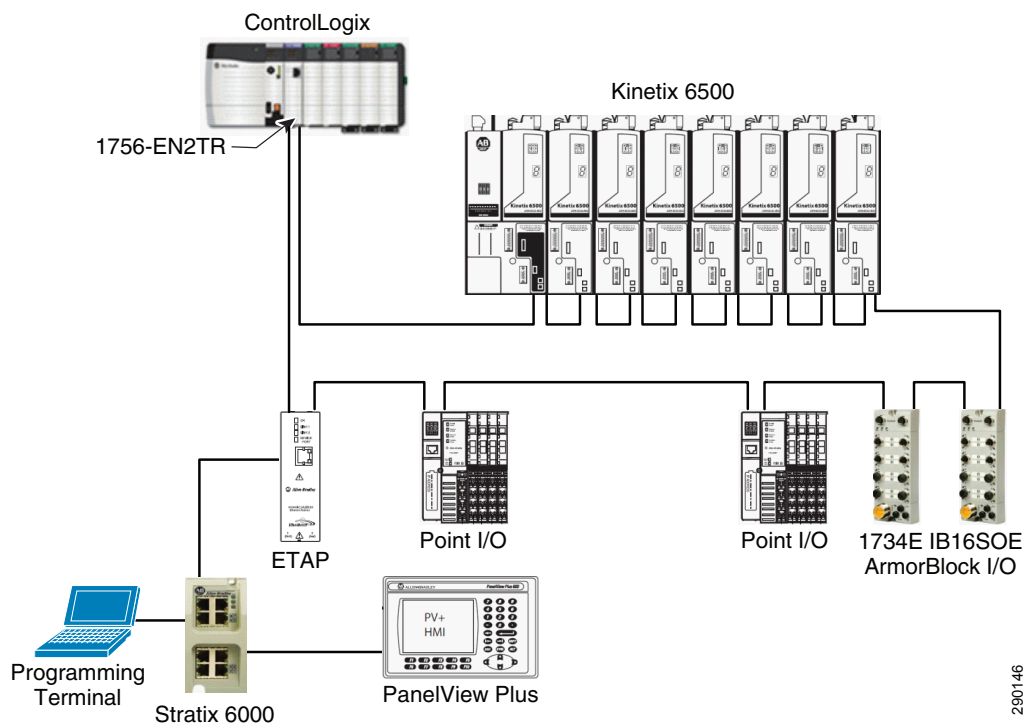
Although the Stratix 8000 supports the transparent clock functionality, it does not support the Beacon ring protocol; therefore, it is not recommended for use in the DLR.

Figure 8-19 DLR Ring Reference Architecture Connected to Stratix 8000 Switch



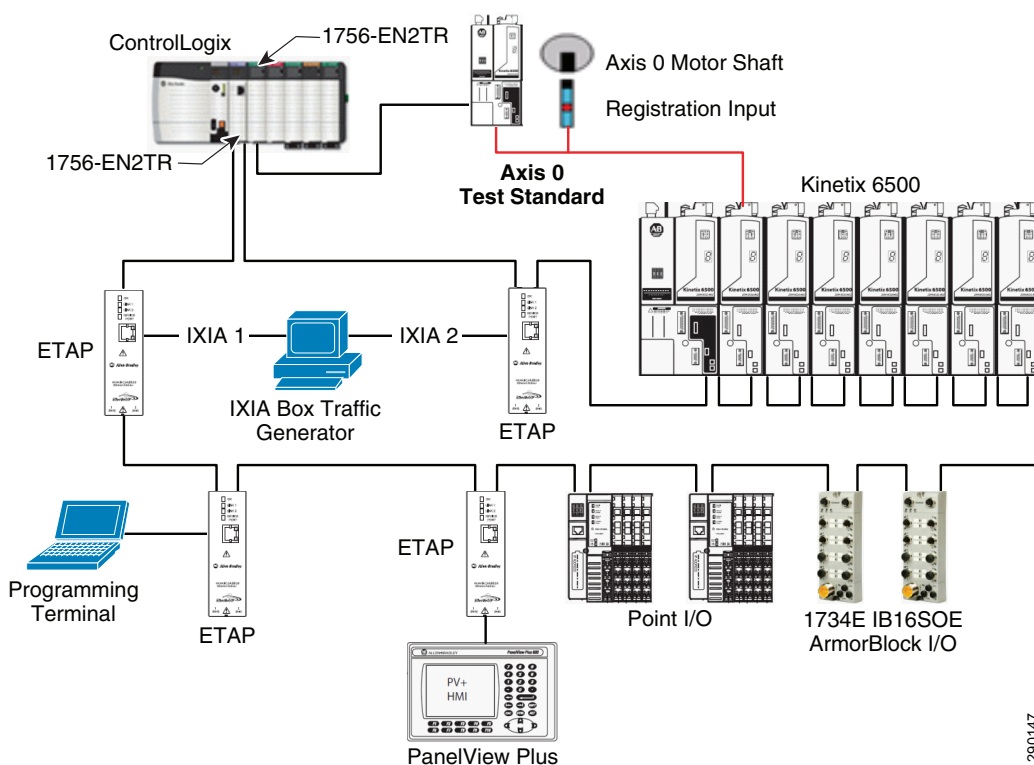
290145

Figure 8-20 DLR Ring Reference Architecture Connected to Stratix 6000 Switch



To test the architecture pictured above, the topology shown in [Figure 8-21](#) was used.

Figure 8-21 DLR Ring Test Architecture



A reference drive (indicated as Axis 0) is used as the reference for all measurements. All measurements, as discussed in [Test Criteria, page 8-29](#), are measured for this architecture.

The Ixia box is a network traffic generator device used to test this architecture. It generates both Class 1 and Class 3 traffic, as well as multicast and unicast traffic on the network. The configuration for the Ixia box is described in [Ixia Network Traffic Generator Configuration, page 8-33](#).

[Table 8-5](#) shows the system configuration parameters for testing this architecture.

Table 8-5 System Configuration Parameters

Controller coarse update rate (ms)	4
Number of CIP Motion axes	8
Number of rack-optimized I/O	2
Number of direct I/O	2
Rack-optimized I/O RPI (ms)	5
Direct I/O RPI (ms)	1
HMI PanelView Plus	1
1783-ETAP	4

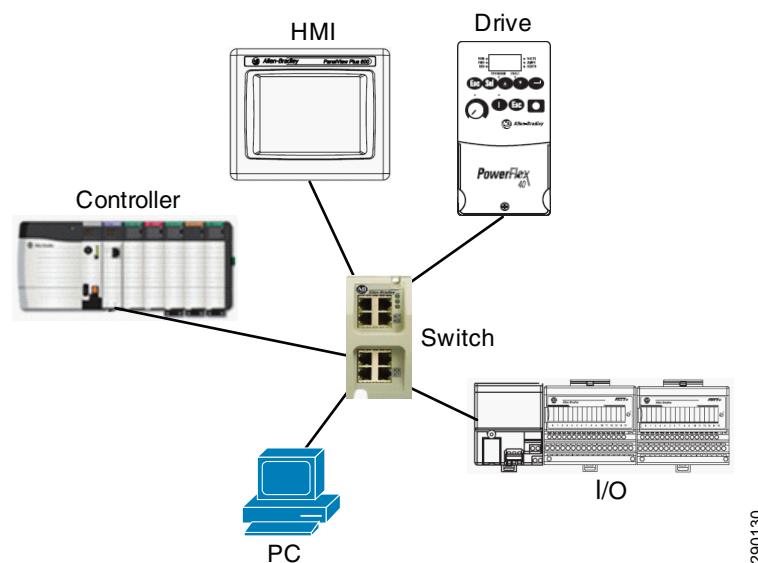
Star Topology

This section discusses various types of star topologies.

The star topology offers the advantage that if a point-to-point connection is lost to an end device, the rest of the network remains intact. The disadvantage of this approach is that all end devices must typically be connected back to a central location. This increases the amount of required cable infrastructure and increases the number of available ports required by the central switch, leading to a higher cost-per-node solution.

In a star network topology, all traffic that traverses the network (that is, device-to-device) must pass through the central switch, as shown in [Figure 8-22](#).

Figure 8-22 Traditional Star Topology

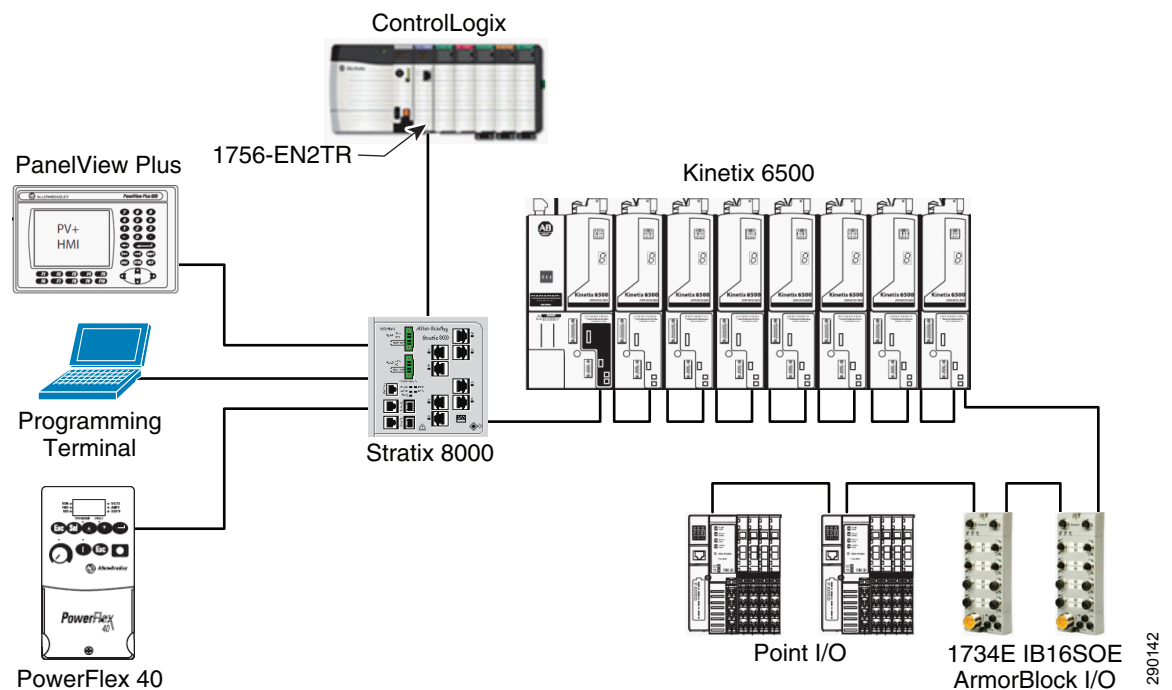


With the advent of devices containing EtherNet/IP embedded switch technology, alternative network topologies can now be achieved, covering a wide range of devices. Embedded switch technology places a multi-port switch directly into end devices, allowing not only for the traditional star, but also linear or ring network topologies. Embedded switch technology has been designed to support the features required by both CIP Sync and CIP Motion.

Star Topology Reference Architectures Under Test

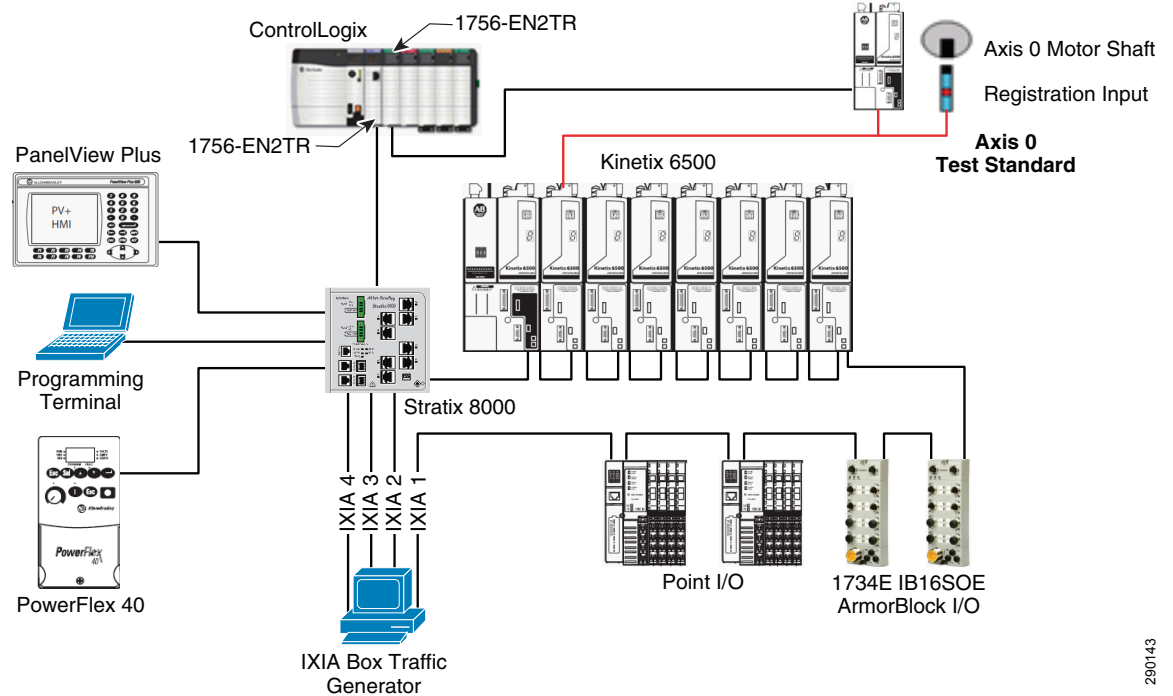
All the components in the architecture are connected in a star topology to the Stratix 8000 Ethernet managed switch, as shown in [Figure 8-23](#). The Stratix 8000 switch has 1588 time synchronization capabilities (transparent and boundary clock).

Figure 8-23 Star Reference Architecture



To test the architecture pictured above, the configuration shown in [Figure 8-24](#) was used.

Figure 8-24 Star Test Architecture



290143

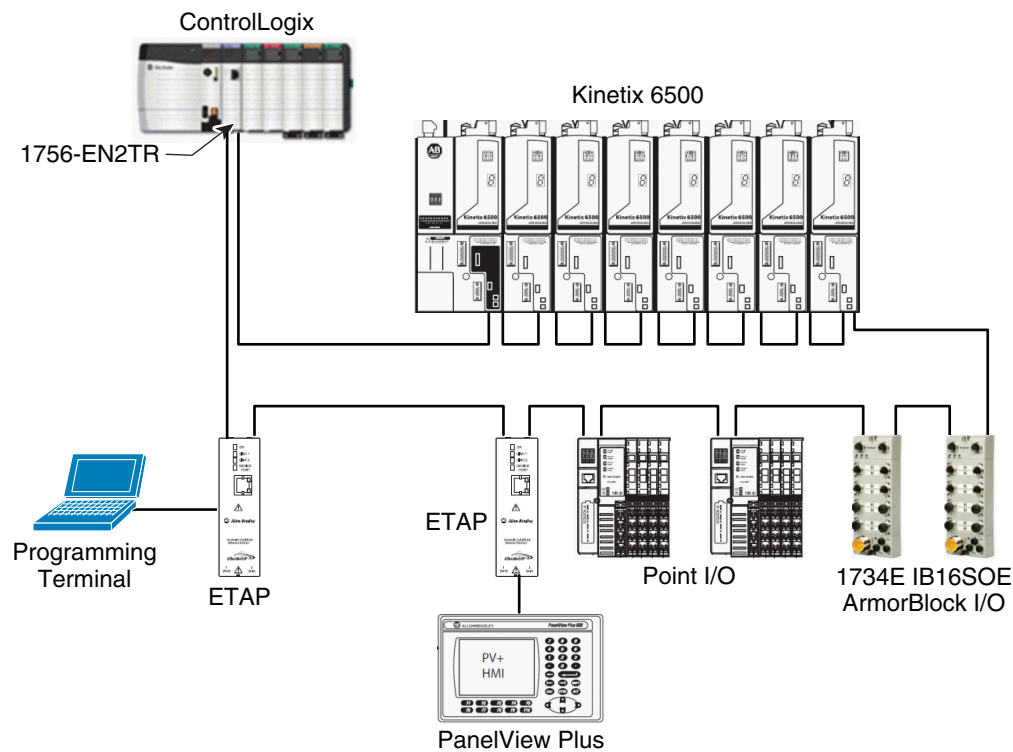
To test the star topology, a reference drive (indicated as Axis 0) is used as the reference for all measurements. All measurements, as discussed in [Test Criteria, page 8-29](#), are measured for this architecture.

The Ixia box is a network traffic generator device used to test this architecture. It generates both Class 1 and Class 3 traffic, as well as multicast and unicast traffic on the network. The configuration for the Ixia box is described in [Ixia Network Traffic Generator Configuration, page 8-33](#).

The Stratix 8000 switch is set up out of the box using Express Setup on the switch. See the documentation accompanying the switch for information on Express Setup.

[Figure 8-25](#) shows the star topology test architecture.

Figure 8-25 Star Topology Test Architecture



The system configuration parameters for testing this architecture are shown in [Table 8-6](#).

Table 8-6 System Configuration Parameters

Controller coarse update rate (ms)	4
Number of CIP Motion axes	8
Number of rack-optimized I/O	2
Number of direct I/O	2
Rack-optimized I/O RPI (ms)	5
Direct I/O RPI (ms)	1
HMI PanelView Plus	1
1783-ETAP	2

CIP Motion Reference Architecture Testing

The goals of the CIP Motion reference architecture testing were as follows:

- Characterize system performance of the CIP Motion system (using Kinetix 6500 drives, PowerFlex 755 drives, and CompactLogix L6x and L7x CIP Motion controllers)
- Validate network performance
- Provide recommended network architectures for Rockwell Automation customers using CIP Motion and CIP Sync

Test Criteria

The basic premise for CIP Motion control is that all devices on the network share a common, precise understanding of time. After time is established on the network, positioning information is sent to each relevant device along with the time that this positioning information is to be acted upon by the device.

To properly test a CIP Motion system, the infrastructure must be stressed in such a way as to attempt to compromise this functionality.

To measure the impact of this disruption in the system, the following three parameters were measured in the connected system:

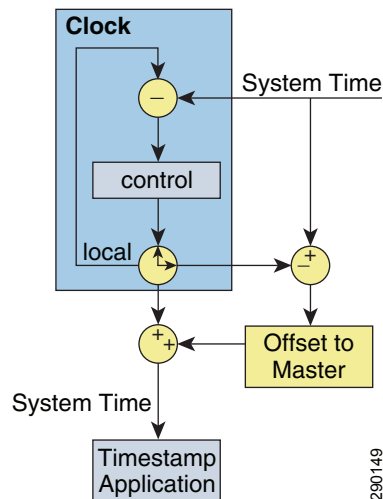
- Offset to master
- Phase error
- Position error

The following sections give an overview of these parameters and how and why they are measured in the system.

Offset to Master

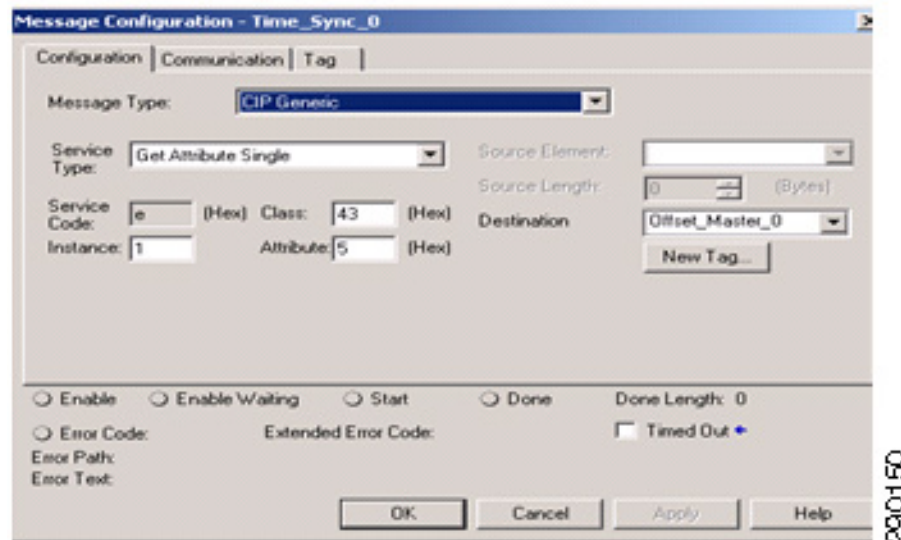
In any system where clock synchronization is accomplished, the local clock of a device powers up with an arbitrary value of time, as compared to system time. In this case, system time is defined as the absolute value of time delivered over the network by the CIP Sync (IEEE 1588) PTP. This value of time is normally referenced to a meaningful value of time as established against Coordinated Universal Time (UTC) time. When system time is delivered over the network, the difference between system time and the local clock time is calculated to create an offset value, referred to as *Offset to Master*, as shown in Figure 8-26. As each device continues to receive new references of time from the grandmaster, this offset value can change, depending on factors such as clock drift, or slight delays in delivery because of traffic or switch utilization.

Figure 8-26 Difference Between System Time and Local Time is the Offset to Master



Each CIP Motion device supports the CIP Sync time sync object. The time sync object supports the Offset to Master attribute. An explicit message instruction (MSG) with the configuration shown in Figure 8-27 is used to measure the Offset to Master value from each drive.

Figure 8-27 Message Configuration for Time Sync Object—Offset to Master



In testing, the change in the Offset to Master value is plotted to reflect the stability of the system and indicate the general health of the time synchronization component of the system. This test is intended to validate the robustness of synchronization accuracy in the face of traffic and loading.

Phase Error

This measurement reflects the phase error between the motion planner and the axes. As the motion planner delivers time and position to the axes, this CIP Motion test measures the effect of data delivery delays because of the network infrastructure or traffic in system. Any delays through the network infrastructure are reflected as an offset or phase error between the actual position at the drive and the commanded position from the controller. It is important to note that this error is seen by all drives, so the drive-to-drive error is virtually zero if the drives share a common infrastructure.

Figure 8-28 illustrates a position phase error.

Figure 8-28 Position Phase Error

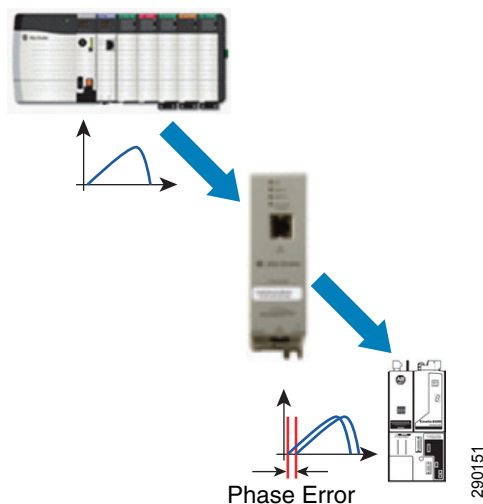


Figure 8-29 and Figure 8-30 show how this measurement is taken. A direct connection is established to a single axis, Axis 0, which is used as the standard for measuring the information coming from the motion planner. This connection goes directly from a dedicated EtherNet/IP module to the Axis 0 drive with no other network components introduced.

All other axes are driven through a network infrastructure. Traffic is injected into the network to introduce traffic loading.

Finally, a registration input is triggered once per revolution and driven into both the Axis 0 and Axis 1 drives. The difference in the respective latched positions reflects the phase error between the two devices. This position error is measured at a constant velocity. To normalize the data independently of velocity, the data is converted to time through the following equation:

$$\text{PhaseErrorTime} = (\text{RegPosAxis0} - \text{RegPosAxis1}) / \text{Velocity}$$

Figure 8-29 Functional Test Arrangement for Phase Error Measurement

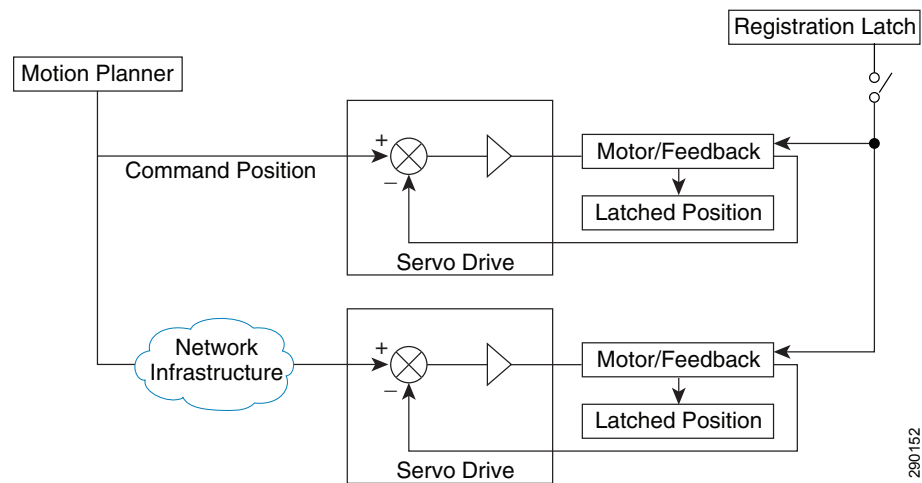
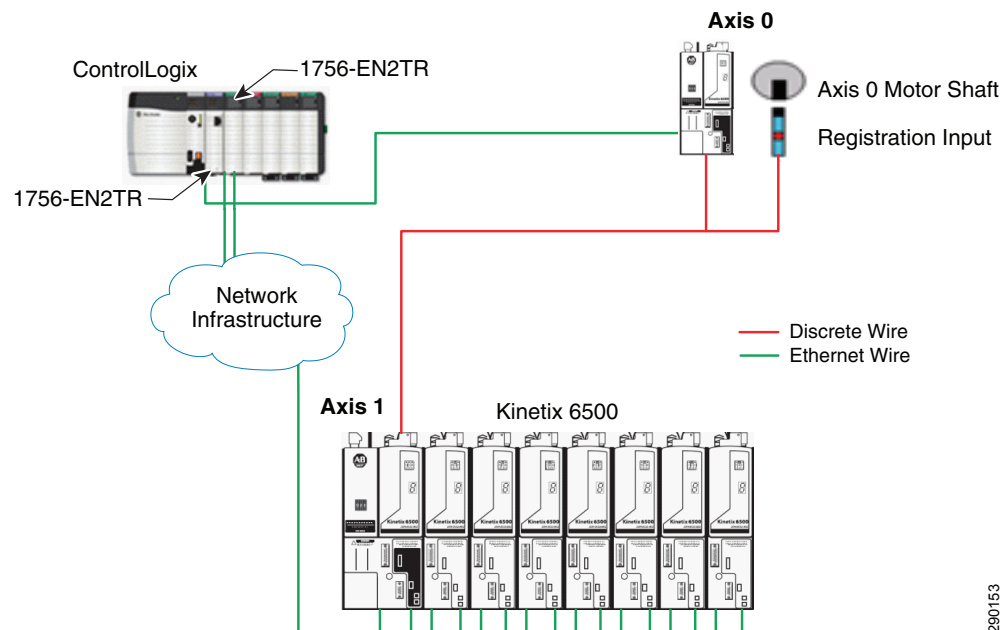


Figure 8-30 Hardware Test Arrangement for Phase Error Measurement



Position Error

This parameter measures position error from the position loop summing junction in the drive during constant velocity regulation. Because the drive receives position and time from the controller, any variation in position error during constant speed regulation is due strictly to clock variation. This parameter, then, measures position error in the drive at constant velocity to determine position error as a function of clock variations.

Figure 8-31 and Figure 8-32 show how this measurement is taken. A direct connection is established to a single axis, Axis 0, which is used as the standard for measuring the information coming from the motion planner. This connection goes directly from a dedicated EtherNet/IP module to the Axis 0 drive with no other network components introduced.

All other axes are driven through a network infrastructure. Traffic is injected into the network to introduce traffic loading.

At every controller motion planner update (coarse update rate), the position error from all the drives is captured. As in the previous measurement, position error is measured at a constant velocity. To normalize the data independently of velocity, the data is converted to time through the following equation:

$$\text{Position Error} = ((\text{Position Error @ Axis}) / \text{Velocity})$$

Figure 8-31 Position Error is Measured from the Position Loop Summing Junction in the Drive

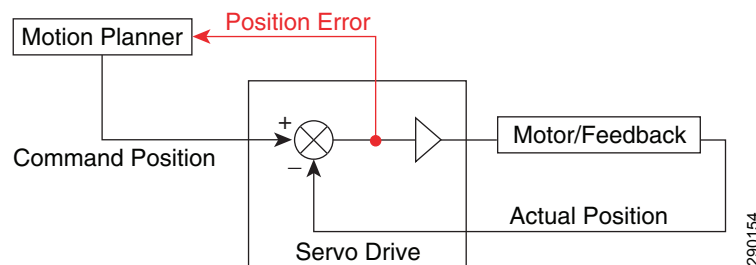
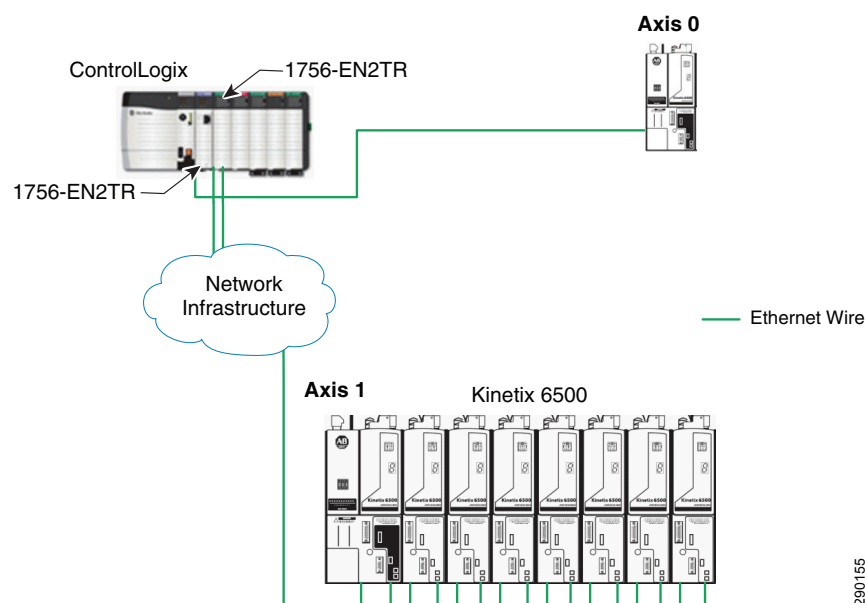


Figure 8-32 Position Error is Captured Every Motion Planner Update



Ixia Network Traffic Generator Configuration

Two ports on the Ixia are used to generate different traffic patterns on the network.

The first port on the Ixia is connected directly to the Stratix 6000 or the Stratix 8000 switch on the network to generate traffic on the switch and stress the switch. This port is referred to as Ixia 3. The TCP/IP traffic stream simulates Layer 3 traffic (MES traffic). The traffic stream generated at Ixia 3 is received at port 4 of Ixia, referred to as Ixia 4.

The second port of the Ixia is connected in the ring or the linear segment, where all the automation components reside. This port is Ixia 2. The traffic stream generated at Ixia 2 is received at port 1 of the Ixia, (Ixia 1). The traffic streams passing through Ixia 1 and Ixia 2 stress the ring or the linear segment. This test is used to estimate the bandwidth capacity on the ring or linear segment.

Three traffic streams are used in the tests. The configuration of these traffic streams is shown in [Table 8-7](#).

Table 8-7 Traffic Stream Configuration

Traffic Pattern	Ixia Source Port	Ixia Destination Port	Traffic Type	Packet Size	Traffic Stream Rate	% of 100 Mbps Capacity
1	2	1	IPv4 TCP/IP—DSCP 58—Class 3 TCP Port 44818	1500 Bytes	700 pps	10%
	1	2	IPv4 TCP/IP—DSCP 58—Class 3	2 Bytes	100 pps	
2	2	1	IPv4 TCP/IP—DSCP 58—Class 3 TCP Port 4872	1500 Bytes	1400 pps	20%
	1	2	IPv4 TCP/IP—DSCP 58—Class 3	2 Bytes	100 pps	
3	2	1	IPv4 TCP/IP—DSCP 58—Class 3 TCP Port 4872	1500 Bytes	2800 pps	30%
	1	2	IPv4 TCP/IP—DSCP 58—Class 3	2 Bytes	100 pps	
4	2	1	IPv4 TCP/IP—DSCP 58—Class 3 TCP Port 4872	1500 Bytes	5600 pps	40%
	1	2	IPv4 TCP/IP—DSCP 58—Class 3	2 Bytes	100 pps	
5	2	1	IPv4 TCP/IP—DSCP 58—Class 3 TCP Port 4872	1500 Bytes	6000 pps	50%
	1	2	IPv4 TCP/IP—DSCP 58—Class 3	2 Bytes	100 pps	
6	2	1	IPv4 TCP/IP—DSCP 58—Class 3 TCP Port 4872	1500 Bytes	8000 pps	60%
	1	2	IPv4 TCP/IP—DSCP 58—Class 3	2 Bytes	100 pps	

Test Results

This section outlines the results of the tests completed for the linear, star, and DLR architectures. These results summarize the tests for these topologies as configured in these test scenarios. The ultimate conclusion for each of these architectures is that the tested loading on each of these architectures has no impact on motion performance. See the results in the tables on the following pages.

See these sections for illustrations of the test architectures:

- [Basic Linear Topologies, page 8-11](#)
- [Linear topology can include many types of devices on the same network, as shown in Figure 8-8., page 8-12](#)
- [Star/Linear Topology, page 8-14](#)
- [Mixed Star/Ring Topology, page 8-20](#)

See the “[Detailed Test Results](#)” section on [page 8-40](#) for a full set of detailed test results.

Linear Architecture

All test parameters previously described were measured in these tests. All the results were measured with respect to the reference Kinetix 6500 drive (Axis 0). The drive is directly connected to the grandmaster clock. No Ixia traffic passes through this reference drive (Axis 0).

The test axis is the Kinetix 6500 drive under test. The Ixia traffic and other automation traffic pass through this drive (Axis 0).

The test traffic patterns are generated using the Ixia box. See the “[Ixia Network Traffic Generator Configuration](#)” section on [page 8-33](#) for more details.

The following tests were performed:

- Test 1.1—Test @ nominal Ixia traffic load
- Test 1.2—Test @ 10% Ixia traffic load
- Test 1.3—Test @ 20% Ixia traffic load
- Test 1.4—Test @ 30% Ixia traffic load
- Test 1.5—Test @ 40% Ixia traffic load
- Test 1.6—Test @ 50% Ixia traffic load

[Table 8-8](#) summarizes the test results.

Table 8-8 Linear Architecture Test Results

Test Case	Test Criterion	Results
Nominal loading	Phase Error:	-315/2.02 μ s
	Position Error:	-1.85/1.79 μ s
	Offset to Master:	-1.93/2.02 μ s
10% loading	Phase Error:	-2.9/2.64 μ s
	Position Error:	-1.96/1.96 μ s
	Offset to Master:	-1.97/2.04 μ s
20% loading	Phase Error:	-2.09/2.76 μ s
	Position Error:	-2.01/1.79 μ s
	Offset to Master:	-1.63/2.06 μ s
30% loading	Phase Error:	-2.09/2.49 μ s
	Position Error:	-2.01/2.07 μ s
	Offset to Master:	-1.89/1.93 μ s

Table 8-8 Linear Architecture Test Results (continued)

Test Case	Test Criterion	Results
40% loading	Phase Error:	-2.45/3.14 μ s
	Position Error:	-1.9/2.07 μ s
	Offset to Master:	-2.06/2.03 μ s
50% loading	Phase Error:	-2.36/3.08 μ s
	Position Error:	-1.96/2.01 μ s
	Offset to Master:	-1.79/1.91 μ s

Star Architecture

All test parameters previously described were measured in these tests. All the results were measured with respect to the reference Kinetix 6500 drive (Axis 0). The drive is directly connected to the grandmaster clock. No Ixia traffic passes through this reference drive (Axis 0).

The following tests were performed:

- Test 2.1—Test @ nominal Ixia traffic load
- Test 2.2—Test @ 10% Ixia traffic load
- Test 2.3—Test @ 20% Ixia traffic load
- Test 2.4—Test @ 30% Ixia traffic load
- Test 2.5—Test @ 40% Ixia traffic load
- Test 2.6—Test @ 50% Ixia traffic load

The test axis is the Kinetix 6500 drive under test. The Ixia traffic and other automation traffic pass through this drive (Axis 0).

The Test Traffic patterns are generated using the Ixia box. See the ["Ixia Network Traffic Generator Configuration" section on page 8-33](#) for more details.

Table 8-9 summarizes the test results.

Table 8-9 Star Architecture Test Results

Test Case	Test Criterion	Results
Nominal loading	Phase Error:	-2.31/2.54 μ s
	Position Error:	-2.23/1.74 μ s
	Offset to Master:	-1.57/1.92 μ s
10% loading	Phase Error:	-2.47/2.04 μ s
	Position Error:	-1.9/1.79 μ s
	Offset to Master:	-1.45/2.05 μ s
20% loading	Phase Error:	-2.41/2.04 μ s
	Position Error:	-2.28/1.74 μ s
	Offset to Master:	-2.06/2.02 μ s
30% loading	Phase Error:	-2.74/2.37 μ s
	Position Error:	-2.17/2.39 μ s
	Offset to Master:	-1.96/2.05 μ s

Table 8-9 Star Architecture Test Results (continued)

Test Case	Test Criterion	Results
40% loading	Phase Error:	-2.23/2.55 μ s
	Position Error:	-2.45/1.74 μ s
	Offset to Master:	-1.78/2.06 μ s
50% loading	Phase Error:	-2.5/2.35 μ s
	Position Error:	-2.23/1.85 μ s
	Offset to Master:	2.08/1.89 μ s

DLR Architecture

All test parameters previously described were measured in these tests. All the results were measured with respect to the reference Kinetix 6500 drive (Axis 0). The drive is directly connected to the grandmaster clock. No Ixia traffic passes through this reference drive (Axis 0).

The test axis is the Kinetix 6500 drive under test. The Ixia traffic and other automation traffic pass through this drive (Axis 0).

The test traffic patterns are generated using the Ixia box. See the [“Ixia Network Traffic Generator Configuration” section on page 8-33](#) for more details.

The following tests were performed:

- Test 0.1—Test @ nominal Ixia traffic load
- Test 0.2—Test @ 10% Ixia traffic load
- Test 0.3—Test @ 20% Ixia traffic load
- Test 0.4—Test @ 30% Ixia traffic load
- Test 0.5—Test @ 40% Ixia traffic load
- Test 0.6—Test @ 50% Ixia traffic load

[Table 8-10](#) summarizes the test results.

Table 8-10 DLR Architecture Test Results

Test Case	Test Criterion	Results
Nominal loading	Phase Error:	-3/2.01 μ s
	Position Error:	-2.17/1.68 μ s
	Offset to Master:	-1.61/ 1.85 μ s
10% loading	Phase Error:	-2.84/2.17 μ s
	Position Error:	-2.28/1.96 μ s
	Offset to Master:	-1.75/ 1.97 μ s
20% loading	Phase Error:	-2.67/2.44 μ s
	Position Error:	-2.45/2.12 μ s
	Offset to Master:	-1.98/ 1.84 μ s
30% loading	Phase Error:	-3/2.4 μ s
	Position Error:	-2.17/1.96 μ s
	Offset to Master:	-1.7/1.99 μ s

Table 8-10 DLR Architecture Test Results (continued)

Test Case	Test Criterion	Results
40% loading	Phase Error:	-2.78/2.27 μ s
	Position Error:	-2.07/2.12 μ s
	Offset to Master:	-1.57/2.04 μ s
50% loading	Phase Error:	-2.56/2.38 μ s
	Position Error:	-2.07/1.74 μ s
	Offset to Master:	-1.53/2.09 μ s

Design Recommendations

Applications that require high accuracy and performance, (for example, high performance motion control) should use devices that support time synchronization and that implement transparent clock or boundary clock mechanisms, such as the following:

- Stratix 8000 switches
- Kinetix 6500 drives
- ArmorBlock I/O
- 1783-ETAP module
- Point I/O
- 1756-ENT2R and 1756-ENT3R modules
- Embedded switch technology
 - Includes transparent clock, Beacon ring protocol, QoS, and IGMP snooping functionality
 - Used in all the devices listed above except the Stratix 8000 switches.

Applications that require less precision and accuracy (for example, general process time-stamping) may not require devices that support boundary or transparent clocks, but clock synchronization is not as accurate. If network components that do not support boundary or transparent clocks are selected, network loading must be kept to less than 20 percent and large packet sizes must be restricted.

These application types can be mixed only on the same subnet as long as those devices that require high precision have a clear view of the system time master via the mechanisms described above (that is, by using devices that maintain time accuracy through the use of transparent or boundary clocks.) This is easily managed in the architecture.

In CIP Motion applications, the use of transparent and boundary clocks, as well as QoS, makes the system extremely robust to variations in network loading.

In this guide, the motion control reference architectures tested were considered “high performance”. In this context, synchronization accuracies of $\sim +2/-2 \mu$ s were observed across the entire system, with phase error lags of $+3/-3 \mu$ s and position error lags of $\sim +2/-2 \mu$ s.

Time Accuracy as a Function of the Application

Motion control is one of many applications that require time synchronization in the control system. In addition to motion control, there are sequence-of-events applications where time stamping is required to determine the order in which certain events occurred. There are data logging applications that use time to associate when data was collected from the system, as well as scheduled-output applications, in which an output can be triggered based on time.

Each of these applications requires different levels of accuracy. Most data logging applications need little more than one-tenth of a second to a second of accuracy when logging data. Motion control, on the other hand, usually requires a much higher degree of synchronization; normally in microseconds (μs) of accuracy.

The CIP Motion reference architectures that are shown in this guide are intended to support high precision motion control applications. It is possible to configure a motion control application with less stringent requirements with an Ethernet infrastructure that does not use time-correcting mechanisms such as transparent clocks and boundary clocks. If this is the case, a different configuration of devices could be incorporated, with less emphasis placed on those components.

To better understand the trade-offs in these applications, consider [Table 8-11](#) and [Figure 8-33](#), [Figure 8-34](#), and [Figure 8-35](#), which show phase error, position error, and Offset to Master when loading an unmanaged switch to 30 percent. In this situation, both phase error and position error were considerably degraded, compared to those architectures that used managed switches with time-compensating tools. In addition, this system could not be loaded beyond 30 percent without dramatically affecting system stability. While unmanaged switches can be used in these applications, care needs to be taken to ensure that traffic loading remains light and that packet sizes are small.

Table 8-11 Unmanaged Switch Without Transparent or Boundary Clock

Test Case	Attribute	Test Result
30% loading	Phase Error	+14/-11 μs
	Position Error	+8/-8 μs
	Offset to Master	+2.3/-2.3 μs

Figure 8-33 Phase Error for Unmanaged Switch Without Transparent or Boundary Clock, 30% Loading

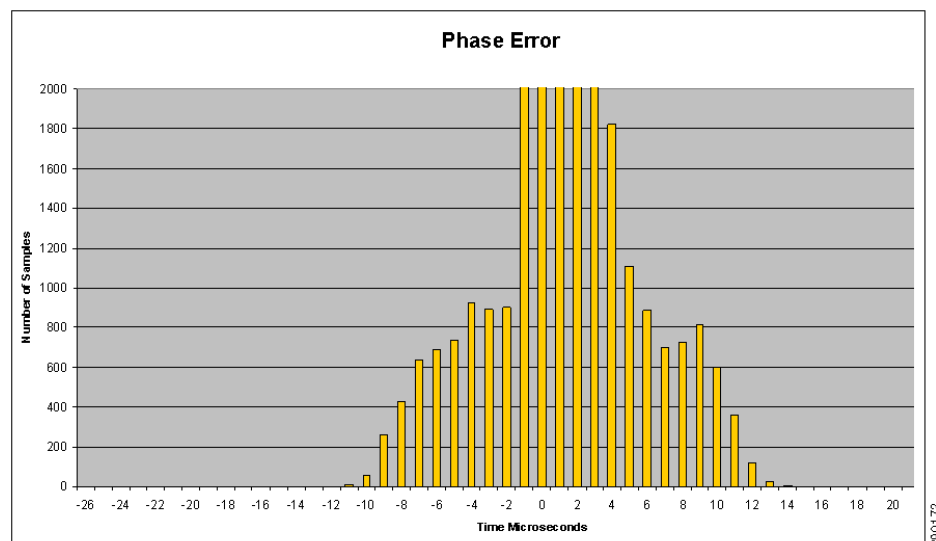


Figure 8-34 Position Error for Unmanaged Switch Without Transparent or Boundary Clock, 30% Loading

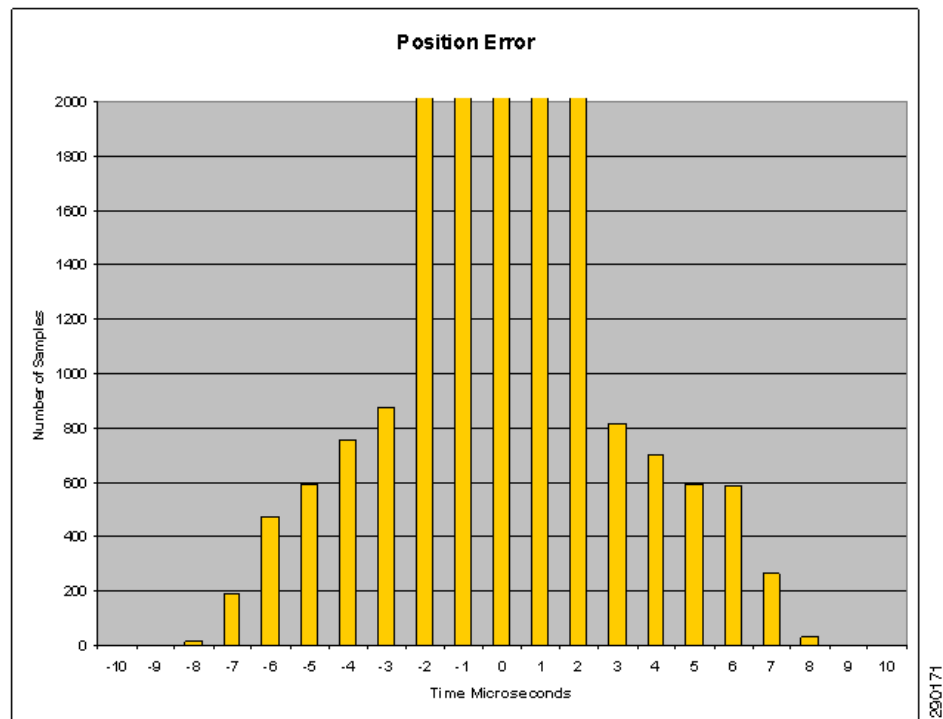
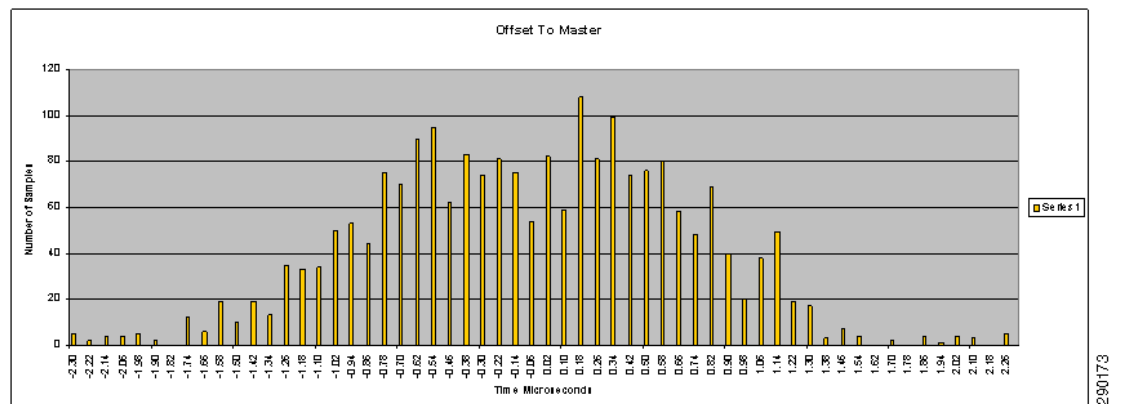


Figure 8-35 Offset to Master for Unmanaged Switch Without Transparent or Boundary clock, 30% Loading



Detailed Test Results

The following is a summary of results for the tests described in this chapter.

Linear Architecture

Table 8-12 and Figure 8-36 through Figure 8-53 summarize the results of the linear architecture test.

Table 8-12 Linear Architecture Test Results

Test 1.1—Test @ Nominal Ixia Traffic Load	
Test Procedure	Step 1Download the RefArchCIPMotion.acd program in the L7controller. Step 2Toggle the Bit Start_Test to start the test. Step 3Collect 100,000 samples. Step 4Open the Excel spreadsheet DataHandling_V7_ArchTest_A8_Linear_BaseTraffic.xls Step 5Click on Connect All Data Sheets to RSLinx Top (Logix Controller) button. Step 6Click on Read All Data from Logix Controller button. Step 7Wait for data to be read. Save the Excel file.
Results Summary	Phase Error: -3.15/2.02 μs Position Error: -1.85/1.79 μs Offset: -1.93/2.02 μs
Test 1.2: Test @ 10 Ixia Traffic Load	
Test Procedure	Step 1Download the RefArchCIPMotion.acd program in the L7controller. Step 2Toggle the Bit Start_Test to start the test. Step 3Collect 100,000 samples. Step 4Open the Excel spreadsheet DataHandling_V7_ArchTest_A8_Linear_10%Traffic.xls Step 5Click on Connect All Data Sheets to RSLinx Top (Logix Controller) button. Step 6Click on Read All Data from Logix Controller button. Step 7Wait for data to be read. Save the Excel file.
Results Summary	Phase Error: -2.9/2.64 μs Position Error: -1.96/1.96 μs Offset: -1.97/2.04 μs
Test 1.3: Test @ 20% Ixia Traffic Load	

Table 8-12 Linear Architecture Test Results (continued)

Test Procedure	Step 1Download the RefArchCIPMotion.acd program in the L7controller. Step 2Toggle the Bit Start_Test to start the test. Step 3Collect 100,000 samples. Step 4Open the Excel spreadsheet DataHandling_V7_ArchTest_A8_Linear_20%Traffic.xls Step 5Click on Connect All Data Sheets to RSLinx Top (Logix Controller) button. Step 6Click on Read All Data from Logix Controller button. Step 7Wait for data to be read. Save the Excel file.
Results Summary	Phase Error: -2.36/3.08 μs Position Error: -1.96/2.01 μs Offset: -1.79/1.91 μs
Test 1.4: Test @ 30% Ixia Traffic Load	
Test Procedure	Step 1Download the RefArchCIPMotion.acd program in the L7controller. Step 2Toggle the Bit Start_Test to start the test. Step 3Collect 100,000 samples. Step 4Open the Excel spreadsheet DataHandling_V7_ArchTest_A8_Linear_30%Traffic.xls Step 5Click on Connect All Data Sheets to RSLinx Top (Logix Controller) button. Step 6Click on Read All Data from Logix Controller button. Step 7Wait for data to be read. Save the Excel file.
Results Summary	Phase Error: -2.09/2.49 μs Position Error: -2.01/2.07 μs Offset: -1.89/1.93 μs
Test 1.5: Test @ 40% Ixia Traffic Load	
Test Procedure	Step 1Download the RefArchCIPMotion.acd program in the L7controller. Step 2Toggle the Bit Start_Test to start the test. Step 3Collect 100,000 samples. Step 4Open the Excel spreadsheet DataHandling_V7_ArchTest_A8_Linear_40%Traffic.xls Step 5Click on Connect All Data Sheets to RSLinx Top (Logix Controller) button. Step 6Click on Read All Data from Logix Controller button. Step 7Wait for data to be read. Save the Excel file.
Results Summary	Phase Error: -2.45/3.14 μs Position Error: -1.9/2.07 μs Offset: -2.06/2.03 μs
Test 1.6: Test @ 50% Ixia Traffic Load	

Table 8-12 Linear Architecture Test Results (continued)

Test Procedure	Step 1Download the RefArchCIPMotion.acd program in the L7controller. Step 2Toggle the Bit Start_Test to start the test. Step 3Collect 100,000 samples. Step 4Open the Excel spreadsheet DataHandling_V7_ArchTest_A8_Linear_50%Traffic.xls Step 5Click on Connect All Data Sheets to RSLinx Top (Logix Controller) button. Step 6Click on Read All Data from Logix Controller button. Step 7Wait for data to be read. Save the Excel file.
Results Summary	Phase Error: -2.09/2.76 μs Position Error: -2.01/1.79 μs Offset: -1.63/2.06 μs

Figure 8-36 Linear Architecture Phase Error Test 1.1—Test @ Nominal Ixia Traffic Load

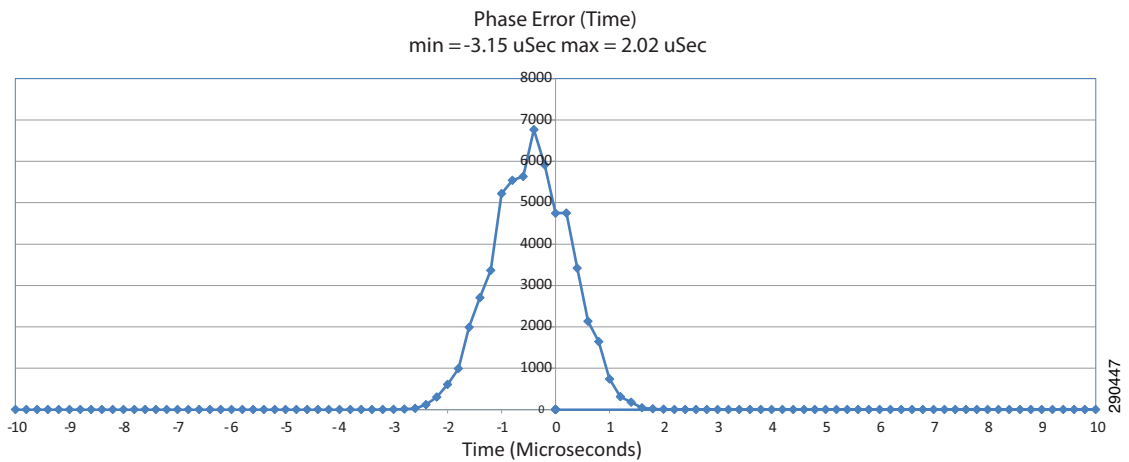


Figure 8-37 Linear Architecture Position Error Test 1.1—Test @ Nominal Ixia Traffic Load

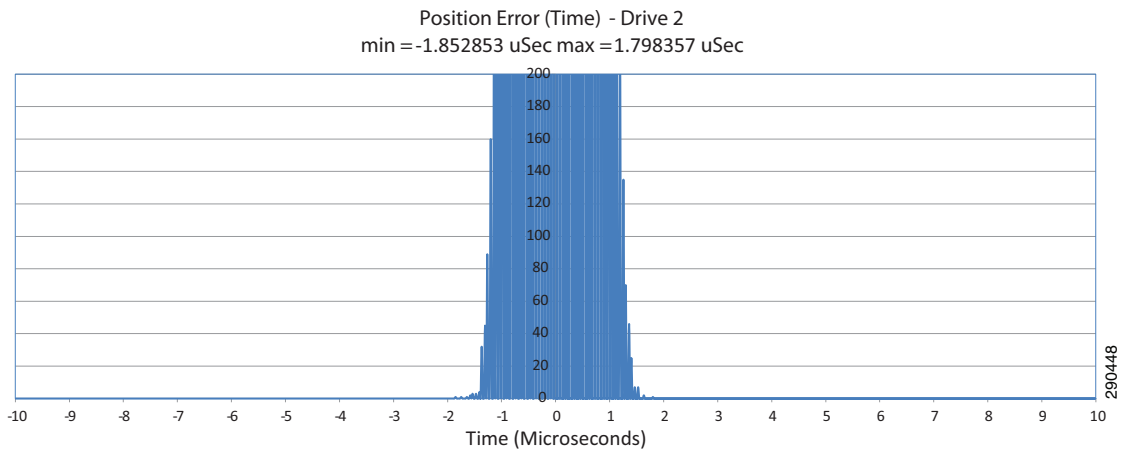


Figure 8-38 Linear Architecture Offset to Master Test 1.1—Test @ Nominal Ixia Traffic Load

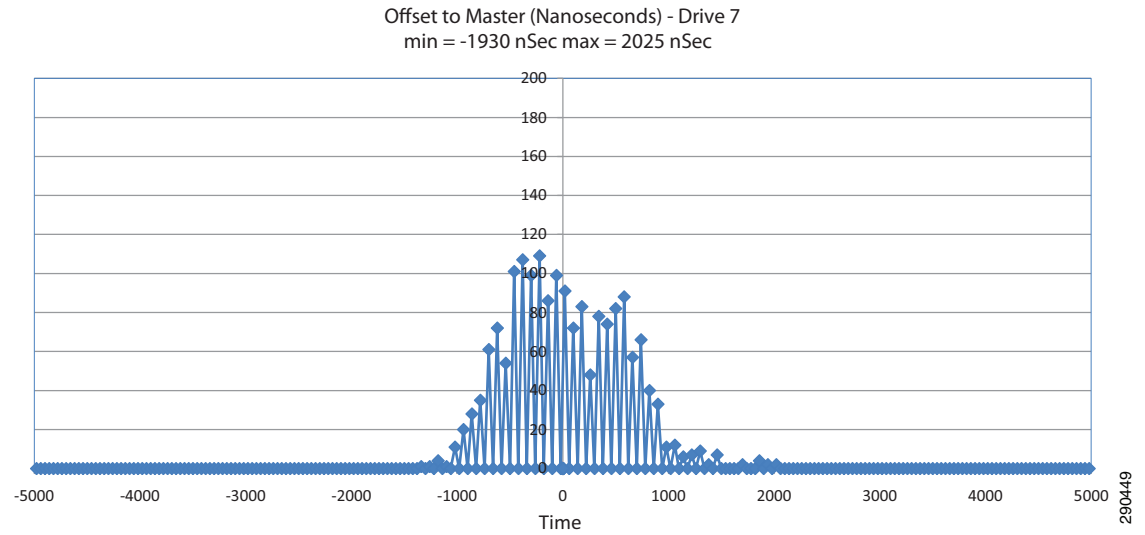


Figure 8-39 Linear Architecture Phase Error Test 1.2—Test @ 10% Ixia Traffic Load

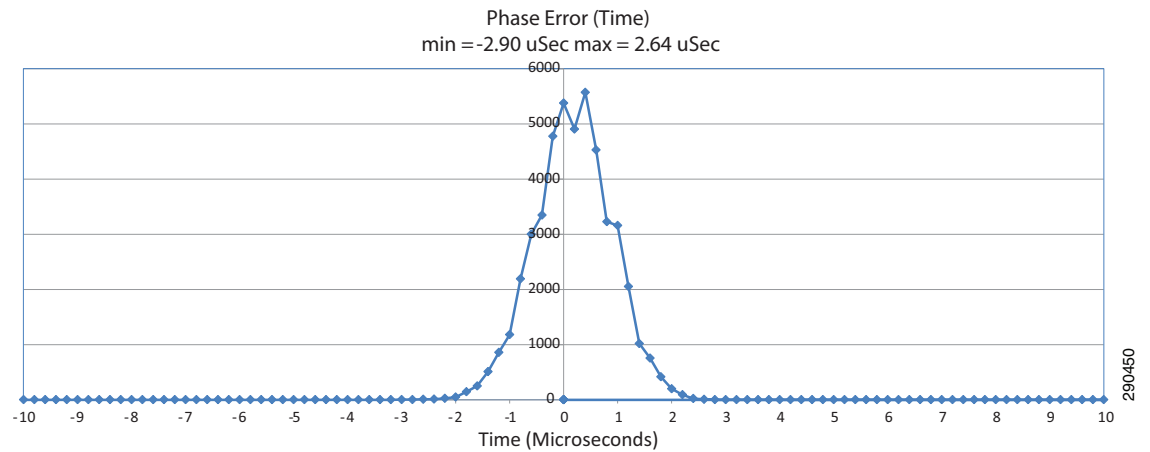


Figure 8-40 Linear Architecture Position Error Test 1.2—Test @ 10% Ixia Traffic Load

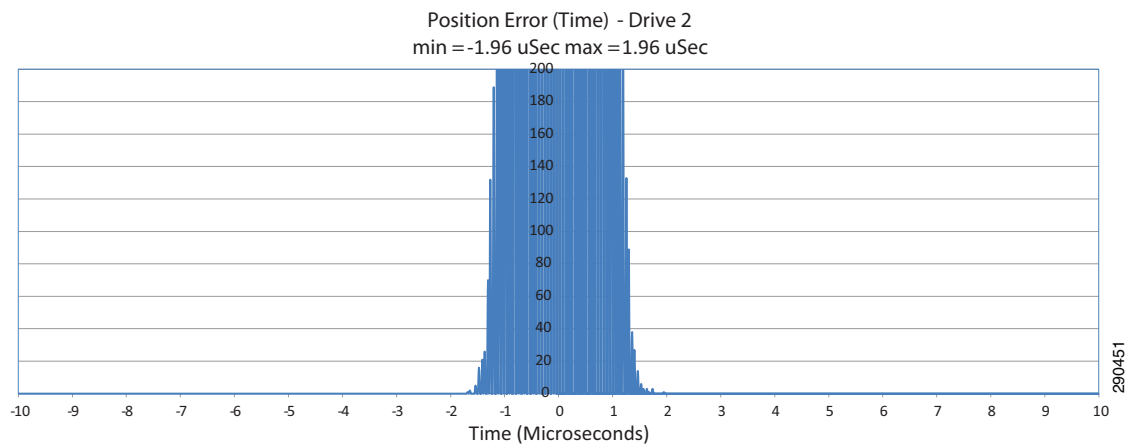


Figure 8-41 Linear Architecture Offset to Master Test 1.2—Test @ 10% Ixia Traffic Load

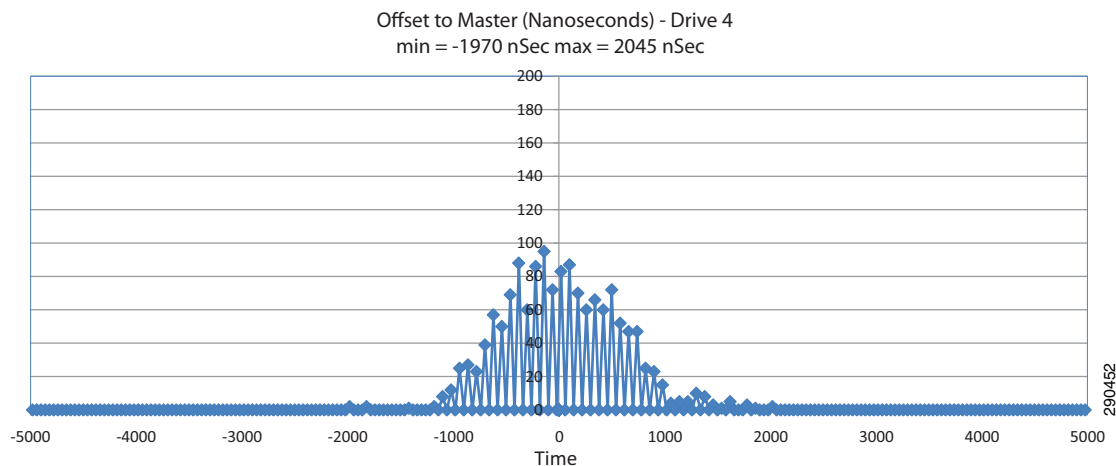


Figure 8-42 Linear Architecture Phase Error Test 1.3—Test @ 20% Ixia Traffic Load

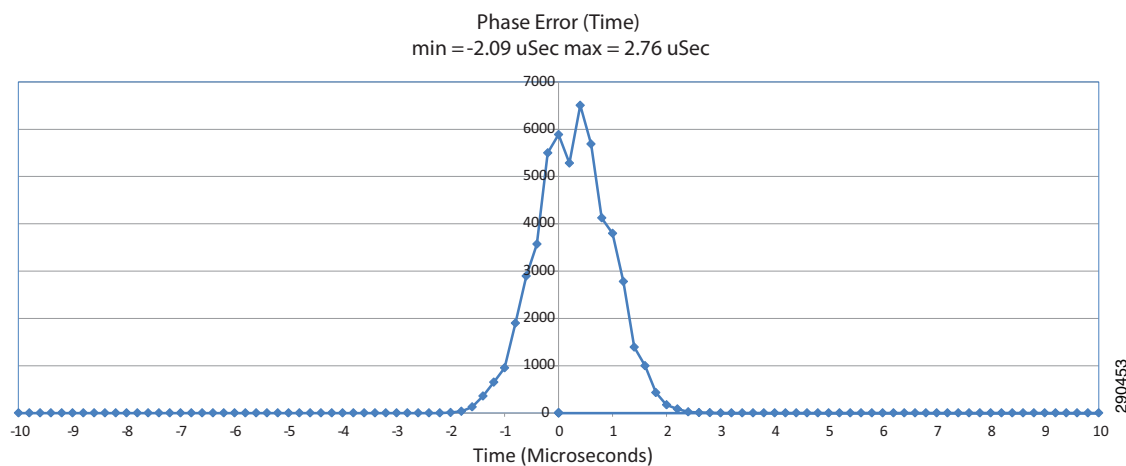


Figure 8-43 Linear Architecture Position Error Test 1.3—Test @ 20% Ixia Traffic Load

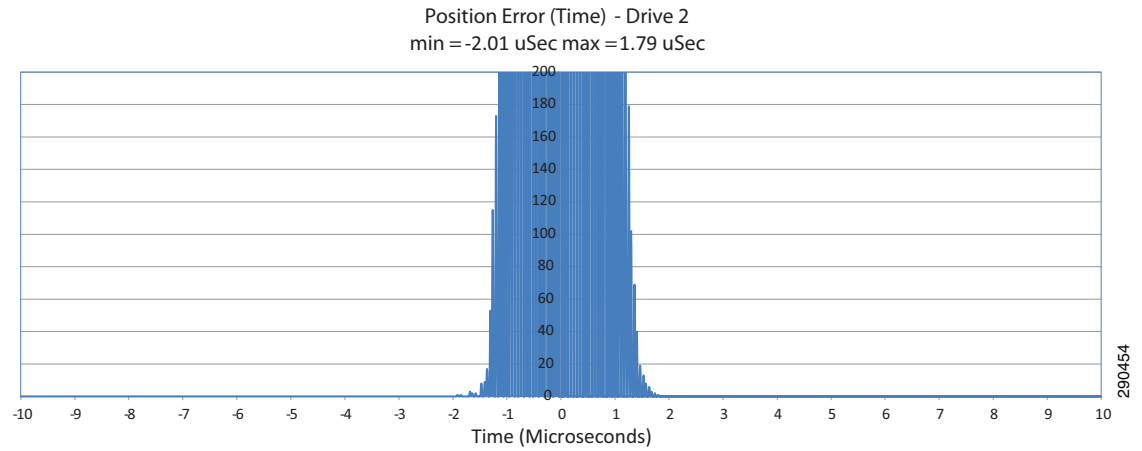


Figure 8-44 Linear Architecture Offset to Master Test 1.3—Test @ 20% Ixia Traffic Load

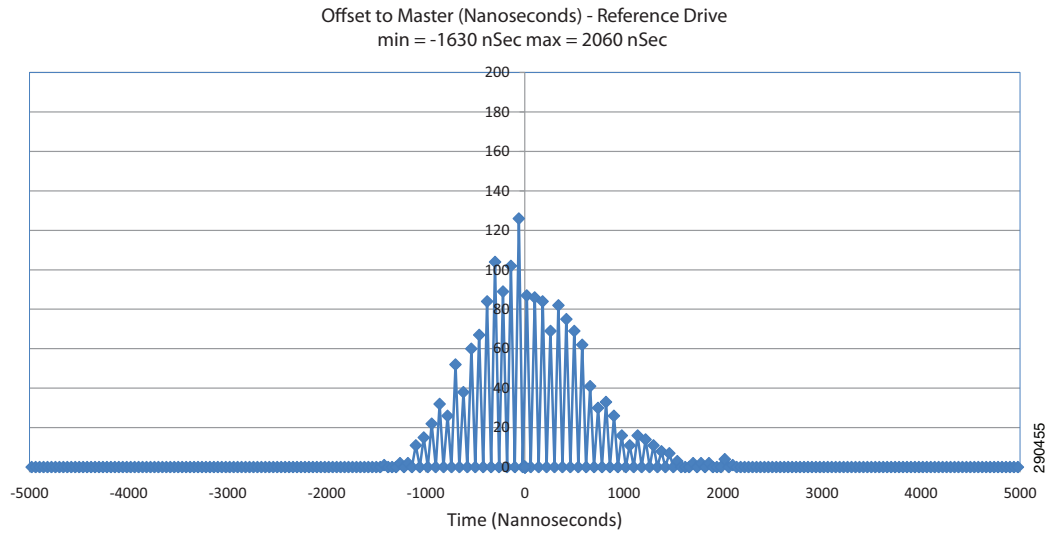


Figure 8-45 Linear Architecture Phase Error Test 1.4—Test @ 30% Ixia Traffic Load

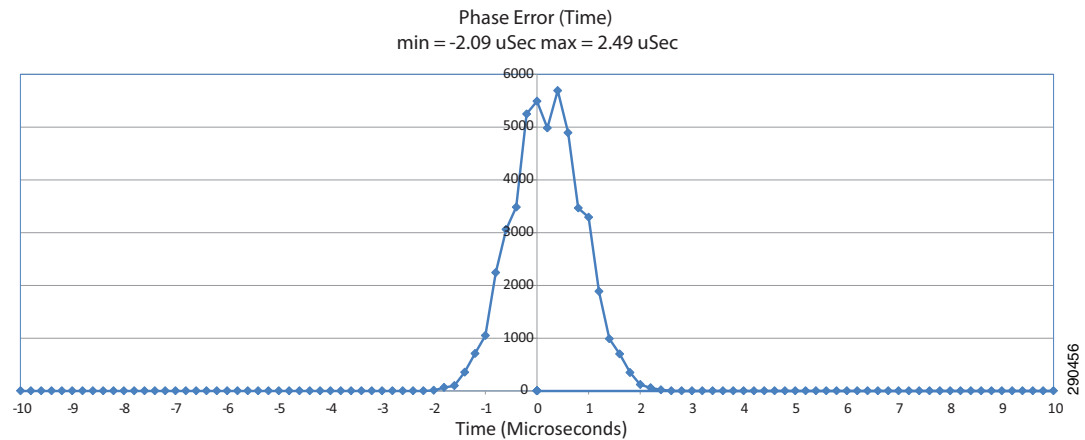


Figure 8-46 Linear Architecture Position Error Test 1.4—Test @ 30% Ixia Traffic Load

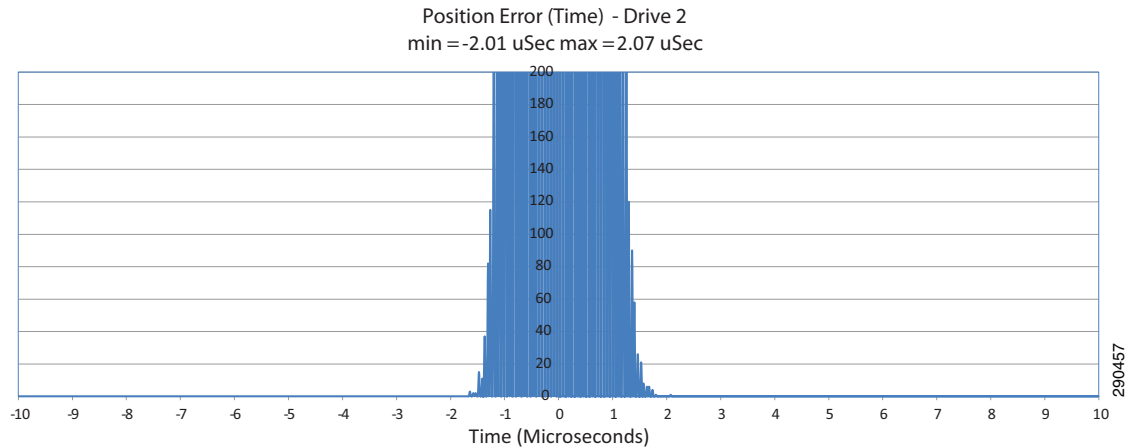


Figure 8-47 Linear Architecture Offset to Master Test 1.4—Test @ 30% Ixia Traffic Load

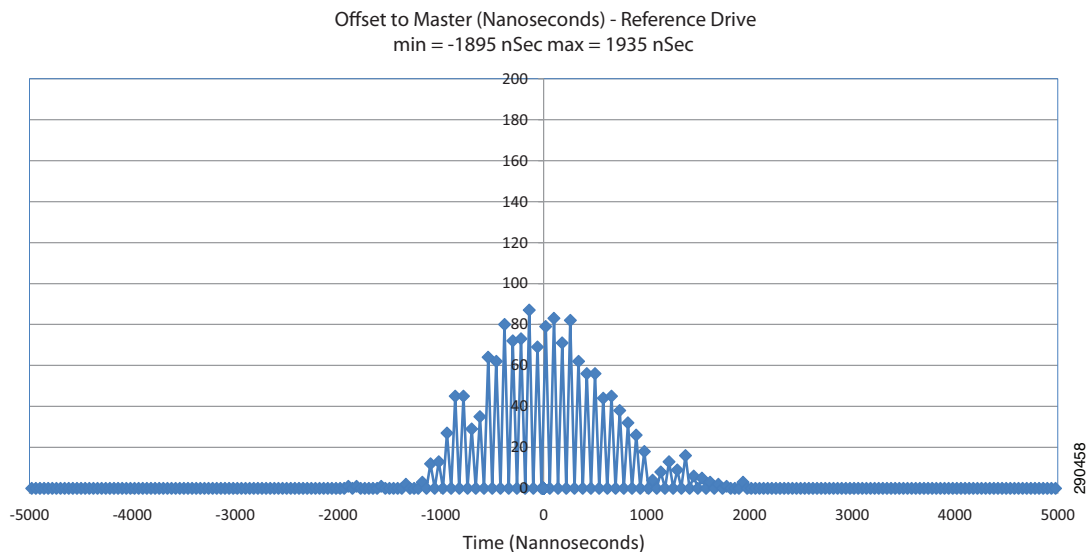


Figure 8-48 Linear Architecture Phase Error Test 1.5—Test @ 40% Ixia Traffic Load

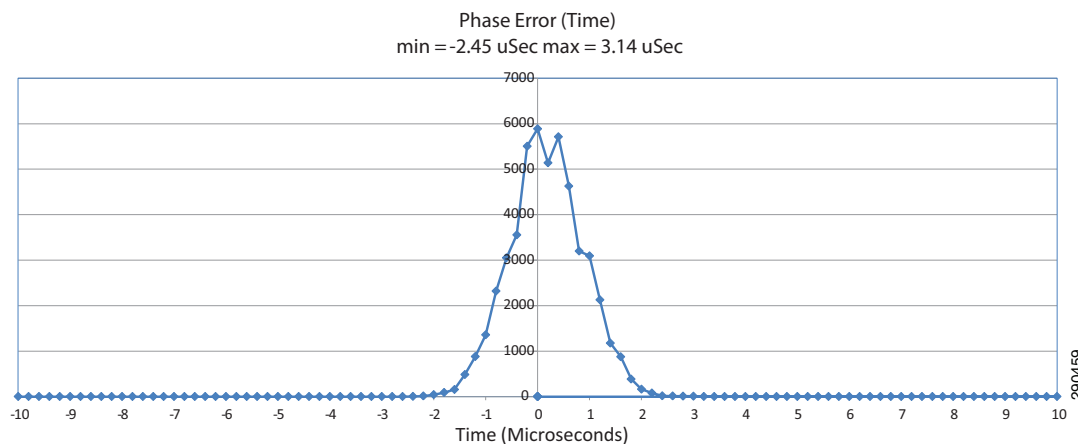


Figure 8-49 Linear Architecture Position Error Test 1.5—Test @ 40% Ixia Traffic Load

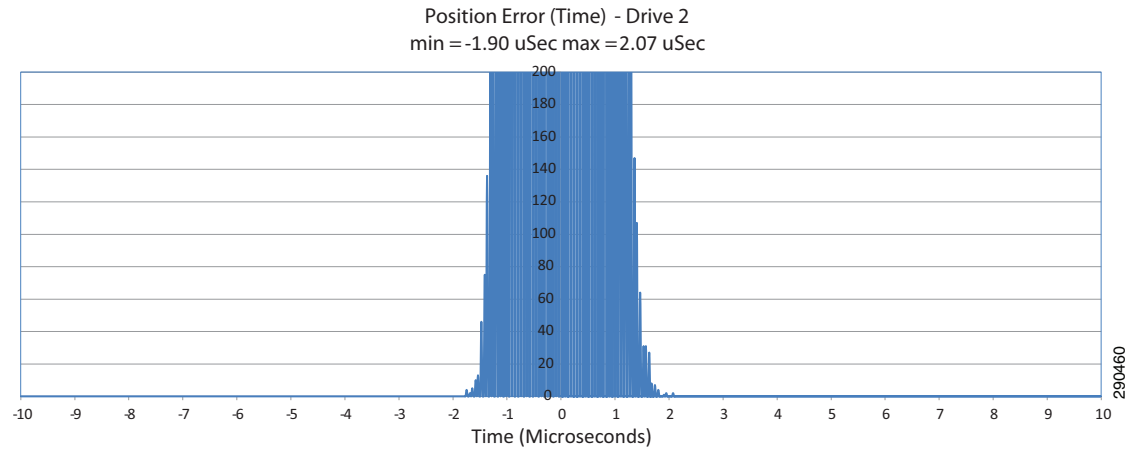


Figure 8-50 Linear Architecture Offset to Master Test 1.5—Test @ 40% Ixia Traffic Load

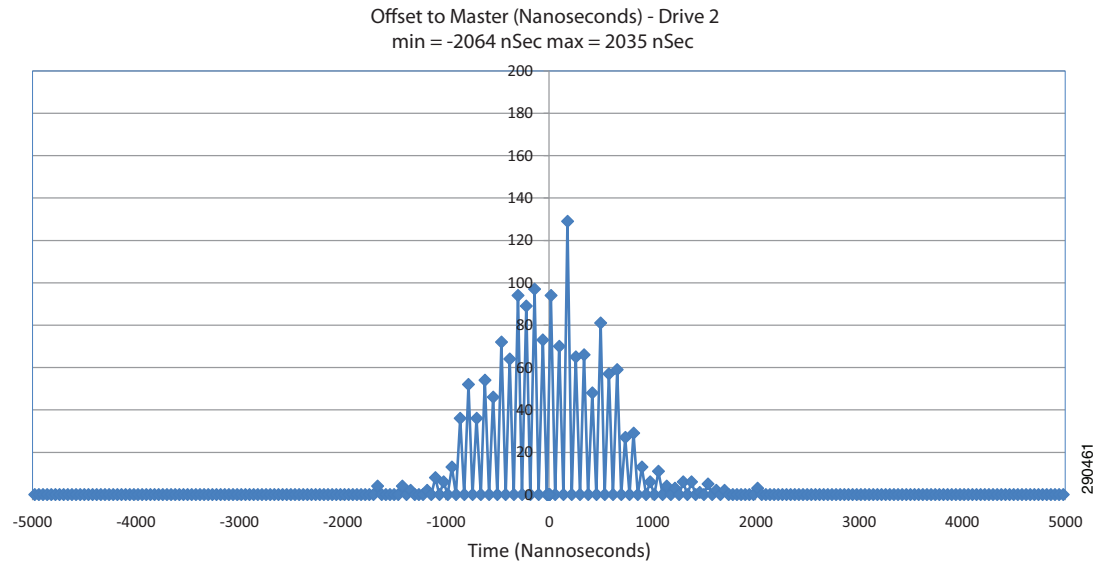


Figure 8-51 Linear Architecture Phase Error Test 1.6—Test @ 50% Ixia Traffic Load

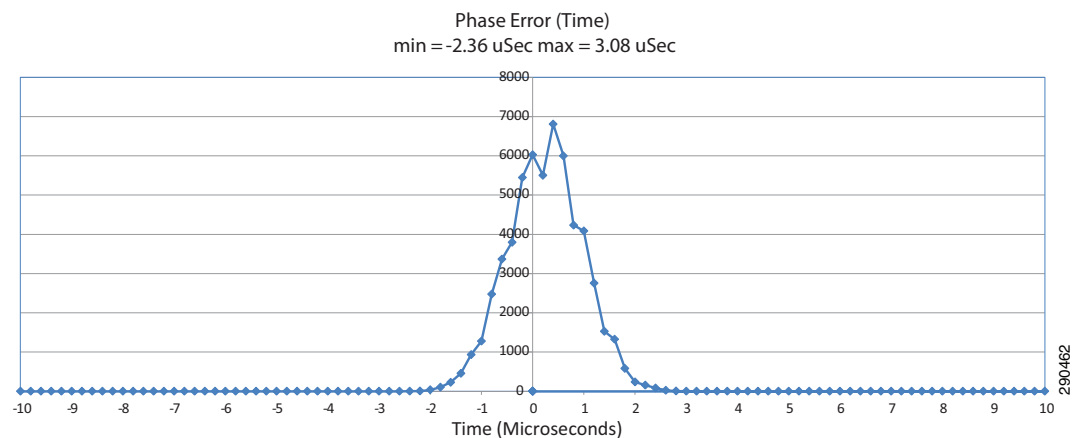


Figure 8-52 Linear Architecture Position Error Test 1.6—Test @ 50% Ixia Traffic Load

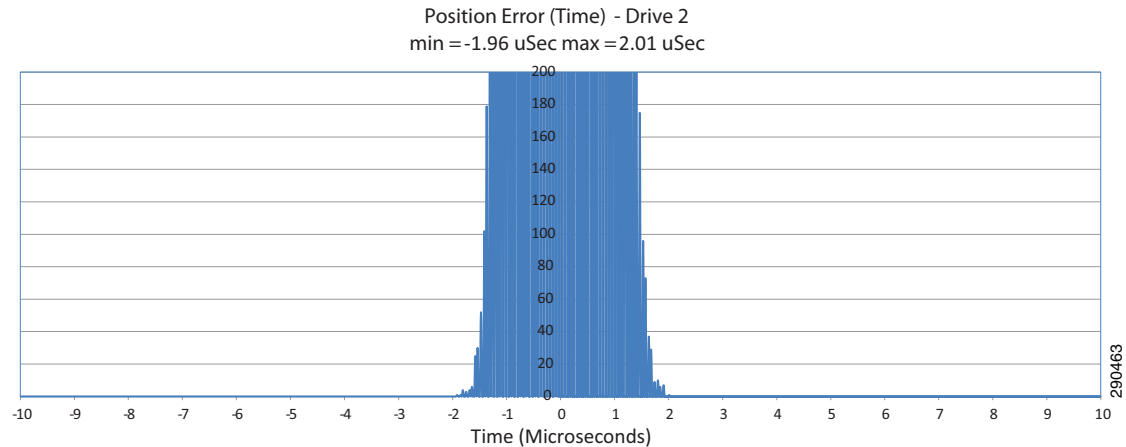
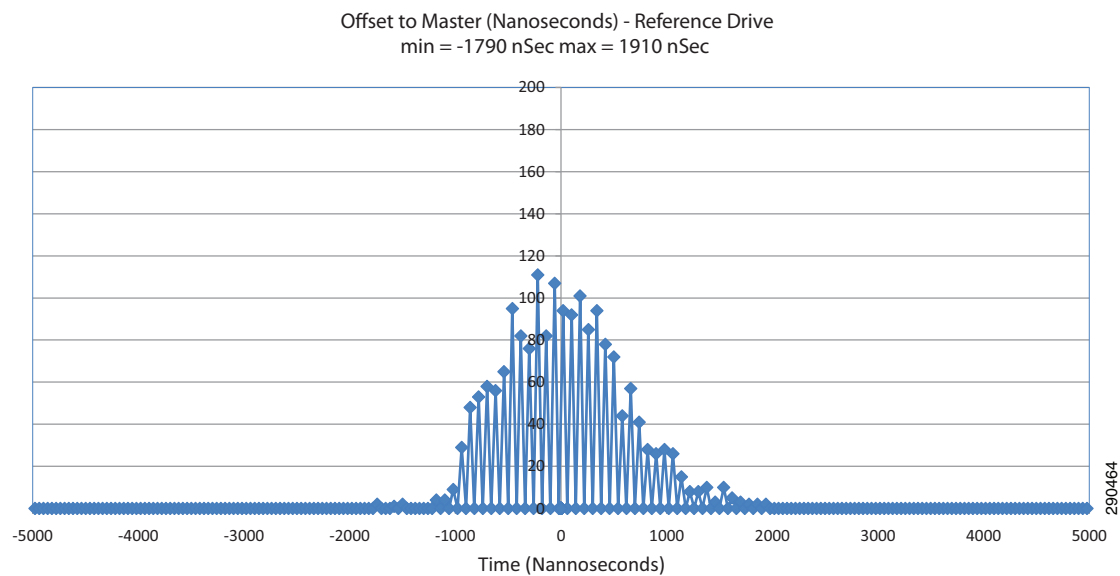


Figure 8-53 Linear Architecture Offset to Master Test 1.6—Test @ 50% Ixia Traffic Load



Star Architecture

Table 8-13 and Figure 8-54 through Figure 8-71 summarize the results of the star architecture test.

Table 8-13 Star Architecture Test Results

Test 2.1 @ Test Nominal IX Traffic Load	
Test Procedure	Step 1Download the RefArchCIPMotion.acd program in the L75controller. Step 2Toggle the Bit Start_Test to start the test. Step 3Collect 100,000 samples. Step 4Open the Excel spreadsheet DataHandling_V7_ArchTest_A8_Linear_BaseTraffic.xls Step 5Click on Connect All Data Sheets to RSLinx Top (Logix Controller) button. Step 6Click on Read All Data from Logix Controller button. Step 7Wait for data to be read. Save the Excel file.
Results Summary	Phase Error: -2.31/2.54 μs Position Error: -2.23/1.74 μs Offset: -1.57/1.92 μs
Test 2.2: Test @ 10% Ixia Traffic Load	
Test Procedure	Step 1Download the RefArchCIPMotion.acd program in the L75controller. Step 2Toggle the Bit Start_Test to start the test. Step 3Collect 100,000 samples. Step 4Open the Excel spreadsheet DataHandling_V7_ArchTest_A8_Linear_BaseTraffic.xls Step 5Click on Connect All Data Sheets to RSLinx Top (Logix Controller) button. Step 6Click on Read All Data from Logix Controller button. Step 7Wait for data to be read. Save the Excel file.
Results Summary	Phase Error: -2.47/2.04 μs Position Error: -1.9/1.79μs Offset: -1.45/2.05 μs
Test 2-3: Test @ 20% Ixia Traffic Load	

Table 8-13 Star Architecture Test Results (continued)

Test Procedure	Step 1Download the RefArchCIPMotion.acd program in the L75controller. Step 2Toggle the Bit Start_Test to start the test. Step 3Collect 100,000 samples. Step 4Open the Excel spreadsheet DataHandling_V7_ArchTest_A8_Linear_BaseTraffic.xls Step 5Click on Connect All Data Sheets to RSLinx Top (Logix Controller) button. Step 6Click on Read All Data from Logix Controller button. Step 7Wait for data to be read. Save the Excel file.
Results Summary	Phase Error: -2.41/2.04 μs Position Error: -2.28/1.74 μs Offset: -2.06/2.02 μs
Test 2.4: Test @ 30% Ixia Traffic Load	
Test Procedure	Step 1Download the RefArchCIPMotion.acd program in the L75controller. Step 2Toggle the Bit Start_Test to start the test. Step 3Collect 100,000 samples. Step 4Open the Excel spreadsheet DataHandling_V7_ArchTest_A8_Linear_BaseTraffic.xls Step 5Click on Connect All Data Sheets to RSLinx Top (Logix Controller) button. Step 6Click on Read All Data from Logix Controller button. Step 7Wait for data to be read. Save the Excel file.
Results Summary	Phase Error: -2.09/2.49 μs Position Error: -2.01/2.07 μs Offset: -1.89/1.93 μs
Test 2.5: Test @ 40% Ixia Traffic Load	
Test Procedure	Step 1Download the RefArchCIPMotion.acd program in the L75controller. Step 2Toggle the Bit Start_Test to start the test. Step 3Collect 100,000 samples. Step 4Open the Excel spreadsheet DataHandling_V7_ArchTest_A8_Linear_BaseTraffic.xls Step 5Click on Connect All Data Sheets to RSLinx Top (Logix Controller) button. Step 6Click on Read All Data from Logix Controller button. Step 7Wait for data to be read. Save the Excel file.

Table 8-13 Star Architecture Test Results (continued)

Results Summary	Phase Error: -2.23/2.55 μ s Position Error: -2.45/1.74 μ s Offset: -1.78/2.06 μ s
Test 2.6: Test @ 50% Ixia Traffic Load	
Test Procedure	Step 1Download the RefArchCIPMotion.acd program in the L75controller. Step 2Toggle the Bit Start_Test to start the test. Step 3Collect 100,000 samples. Step 4Open the Excel spreadsheet DataHandling_V7_ArchTest_A8_Linear_BaseTraffic.xls Step 5Click on Connect All Data Sheets to RSLinx Top (Logix Controller) button. Step 6Click on Read All Data from Logix Controller button. Step 7Wait for data to be read. Save the Excel file.
Results Summary	Phase Error: -2.5/2.35 μ s Position Error: -2.23/1.85 μ s Offset: 2.08/1.89 μ s

Figure 8-54 Star Architecture Phase Error Test 2.1—Test @ Nominal Ixia Traffic Load

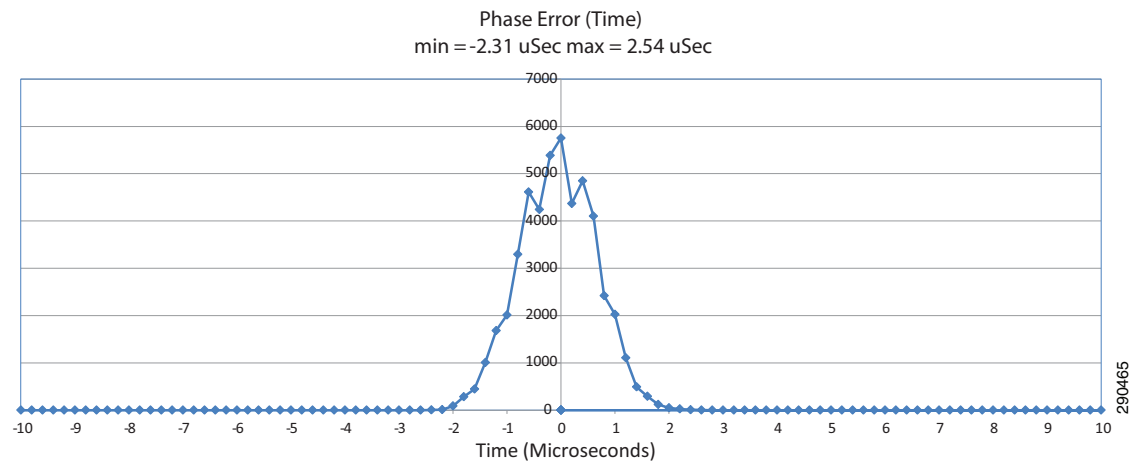


Figure 8-55 Star Architecture Position Error Test 2.1—Test @ Nominal Ixia Traffic Load

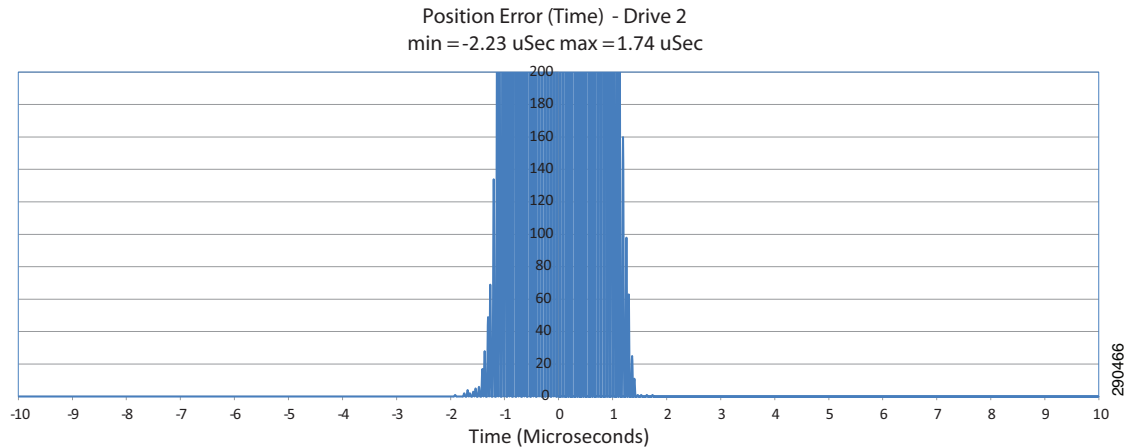


Figure 8-56 Star Architecture Offset to Master Test 2.1—Test @ Nominal Ixia Traffic Load

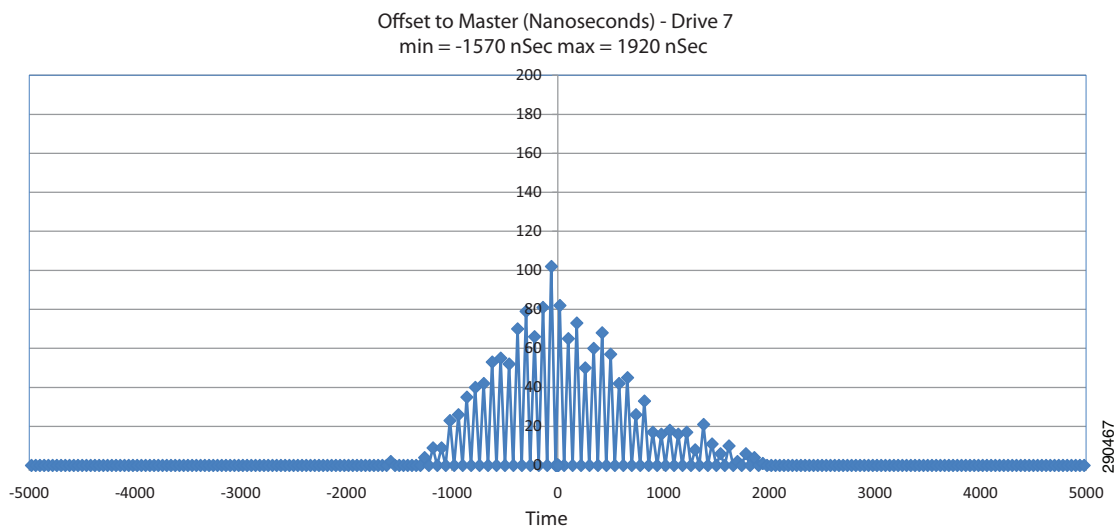


Figure 8-57 Star Architecture Phase Error Test 2.2—Test @ 10% Ixia Traffic Load

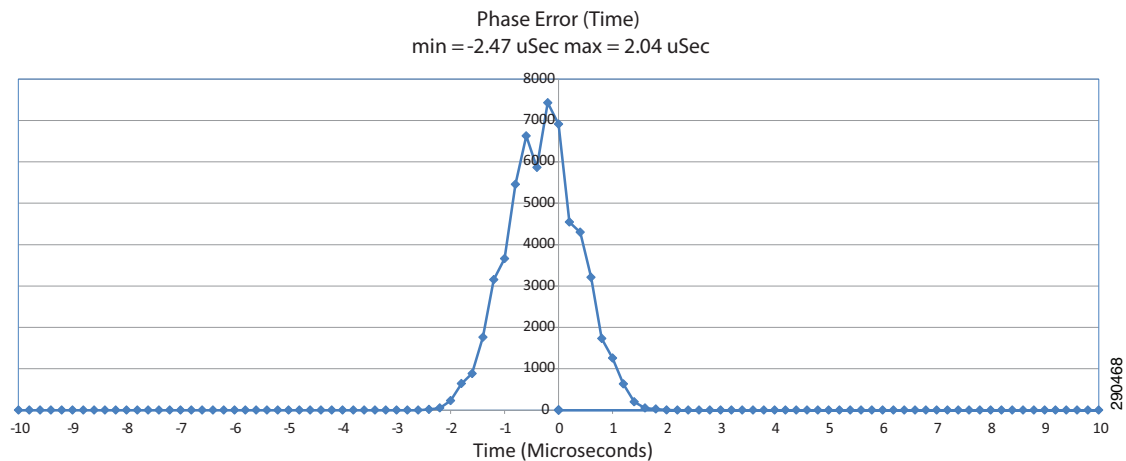


Figure 8-58 Star Architecture Position Error Test 2.2—Test @ 10% Ixia Traffic Load

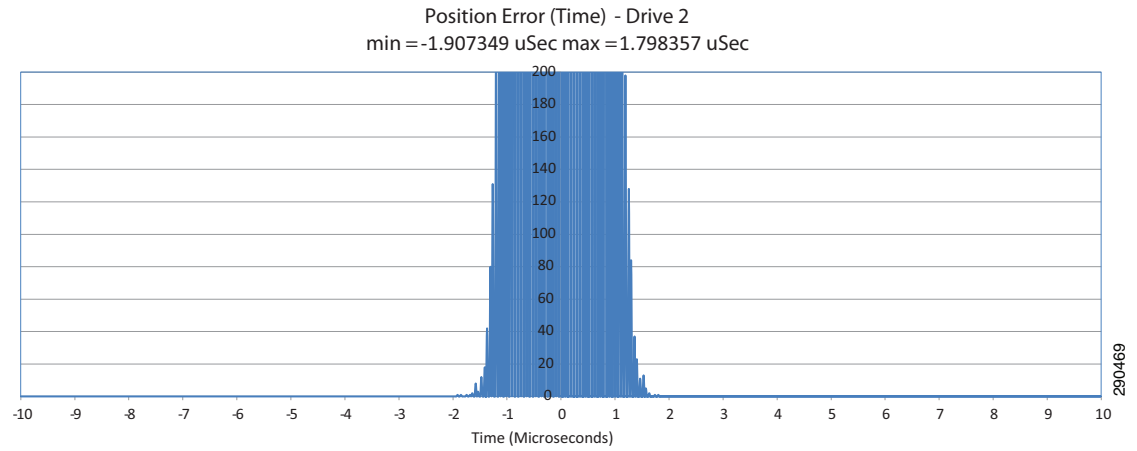


Figure 8-59 Star Architecture Offset to Master Test 2.2—Test @ 10% Ixia Traffic Load

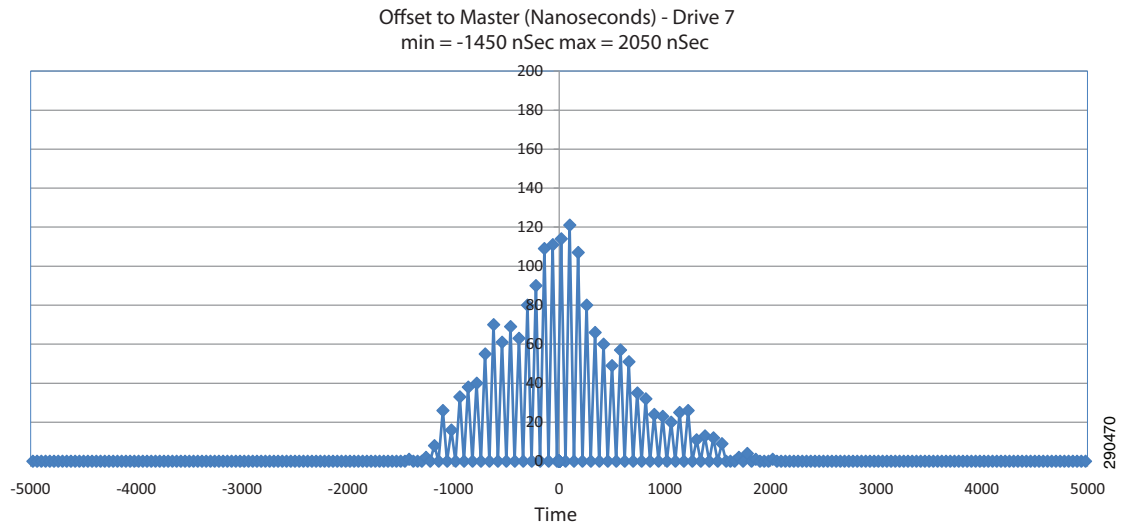


Figure 8-60 Star Architecture Phase Error Test 23—Test @ 20% IX.IA Traffic Load

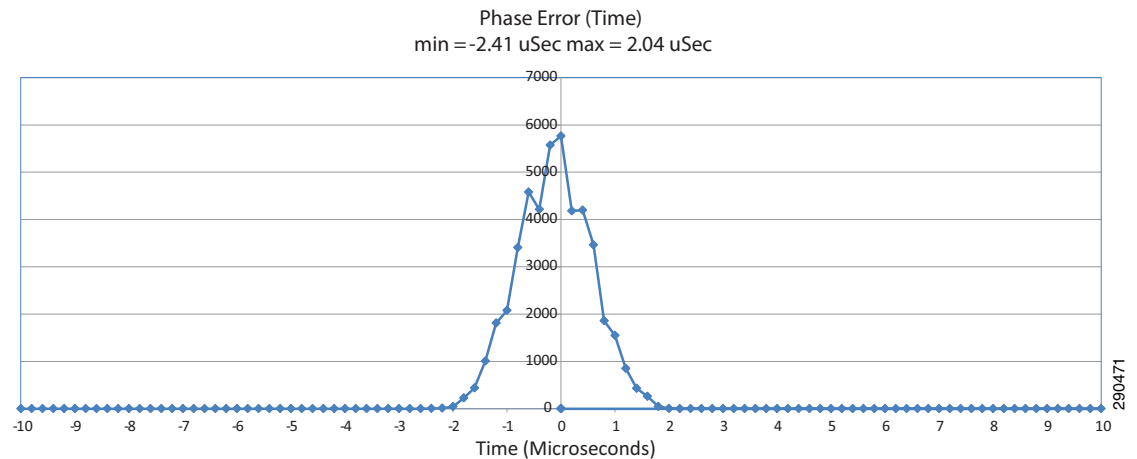


Figure 8-61 Star Architecture Position Error Test 2.3—Test @ 20% Ixia Traffic Load

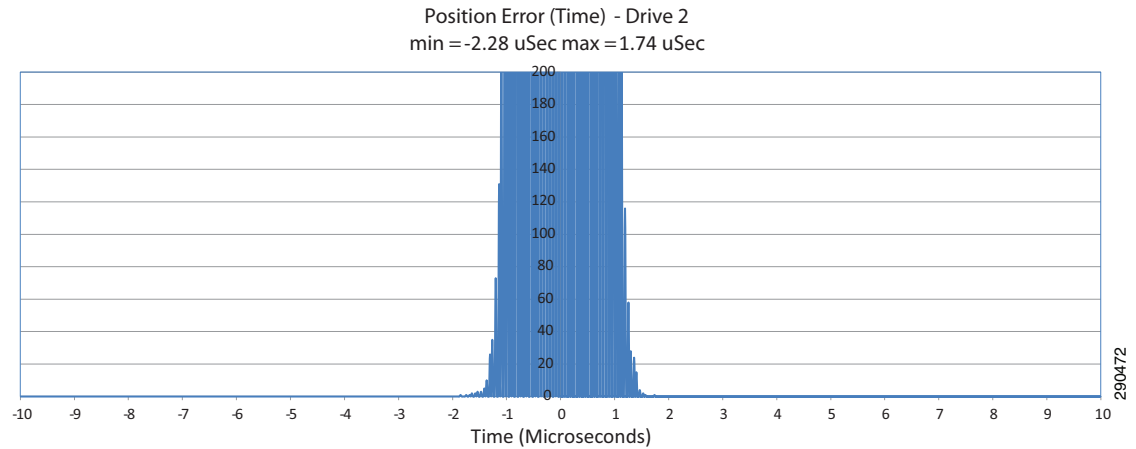


Figure 8-62 Star Architecture Offset to Master Test 2.3—Test @ 20% Ixia Traffic Load

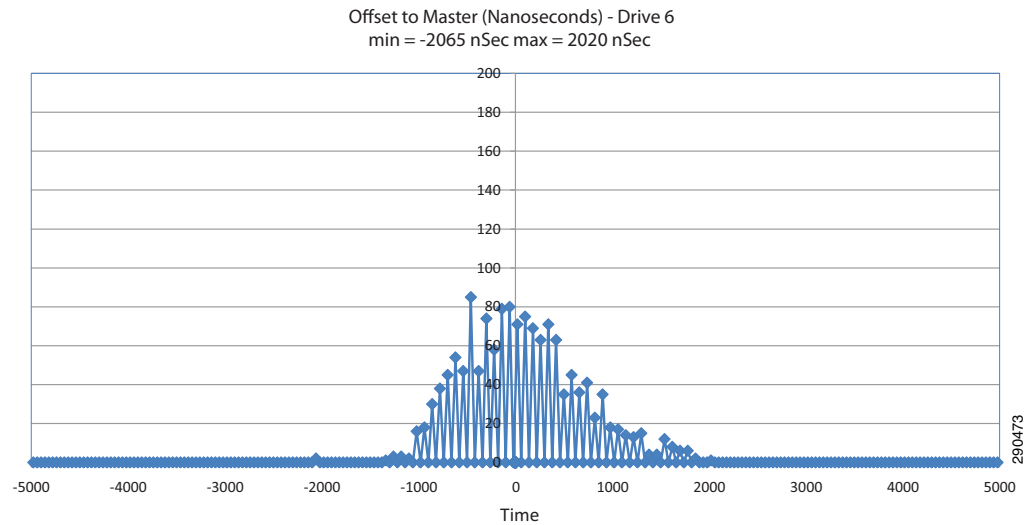


Figure 8-63 Star Architecture Phase Error Test 2.4—Test @ 30% Ixia Traffic Load

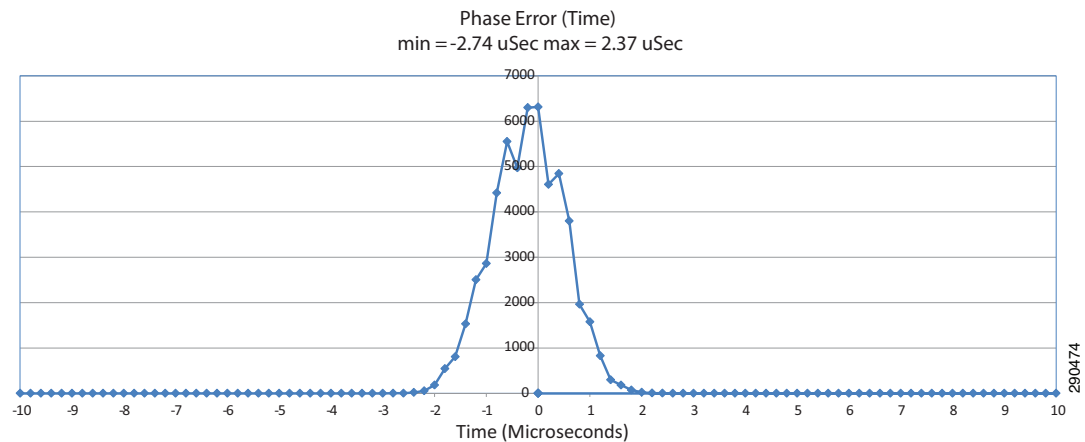


Figure 8-64 Star Architecture Position Error Test 2.4—Test @ 30% Ixia Traffic Load

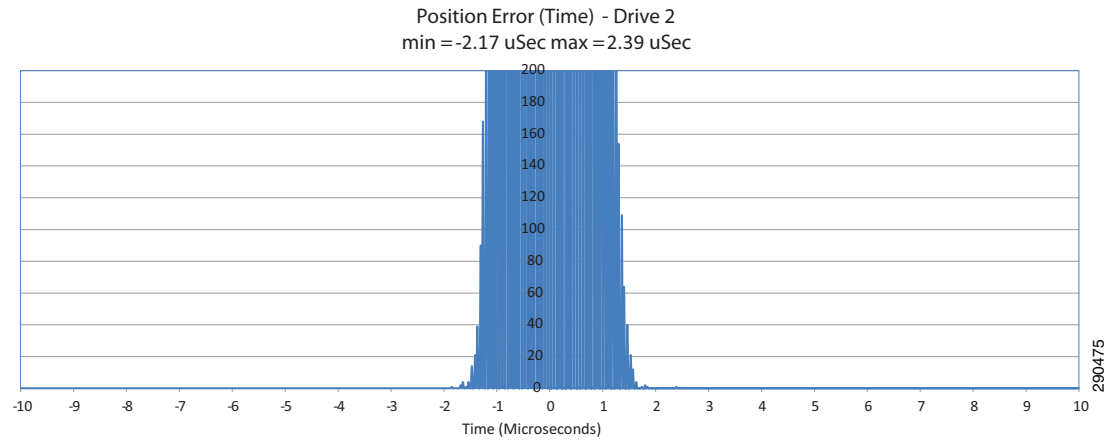


Figure 8-65 Star Architecture Offset to Master Test 2.4—Test @ 30% Ixia Traffic Load

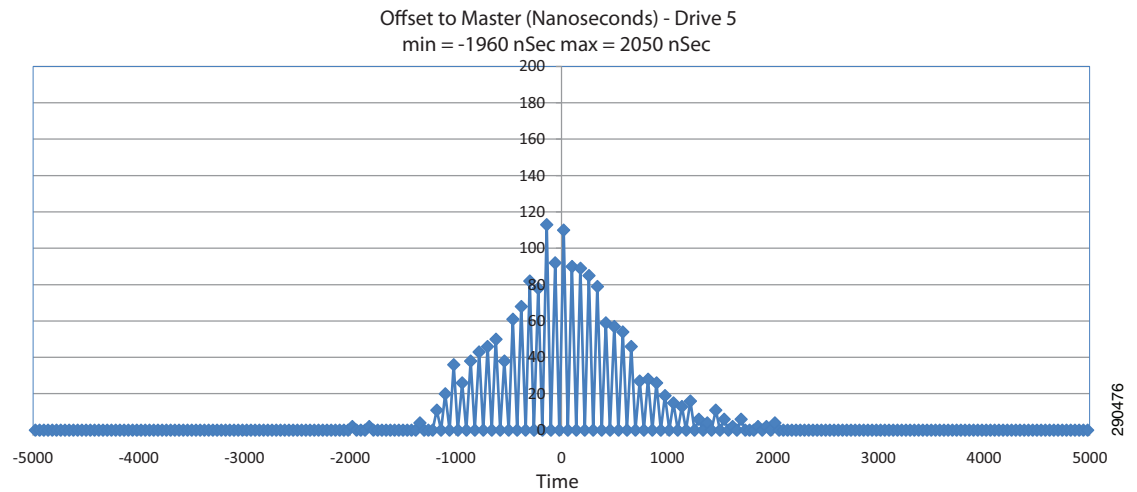


Figure 8-66 Star Architecture Phase Error Test 2.5—Test @ 40% Ixia Traffic Load

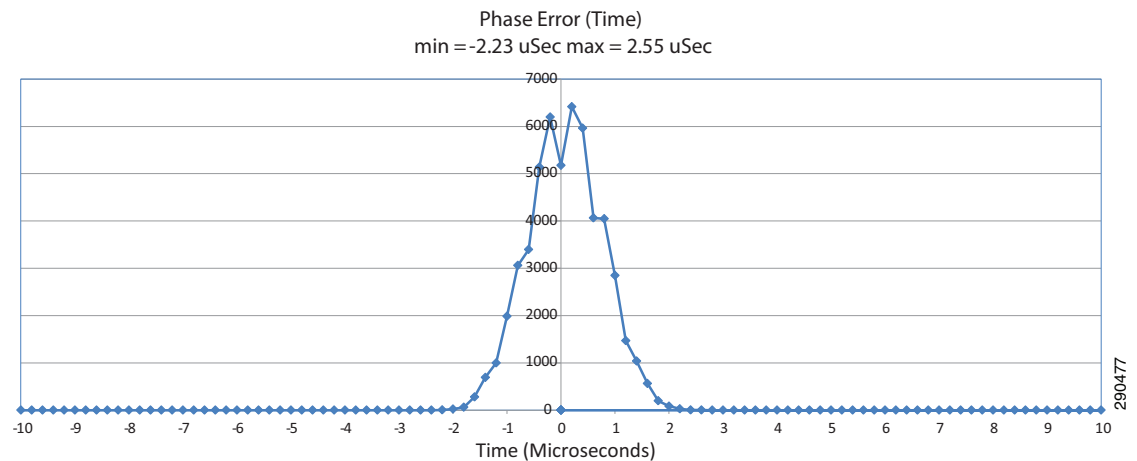


Figure 8-67 Star Architecture Position Error Test 2.5—Test @ 40% Ixia Traffic Load

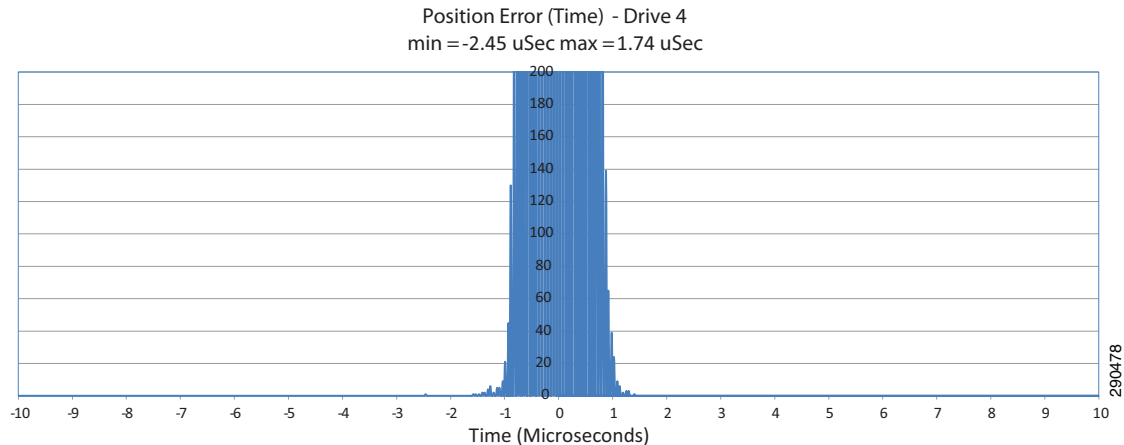


Figure 8-68 Star Architecture Offset to Master Test 2.5—Test @ 40% Ixia Traffic Load

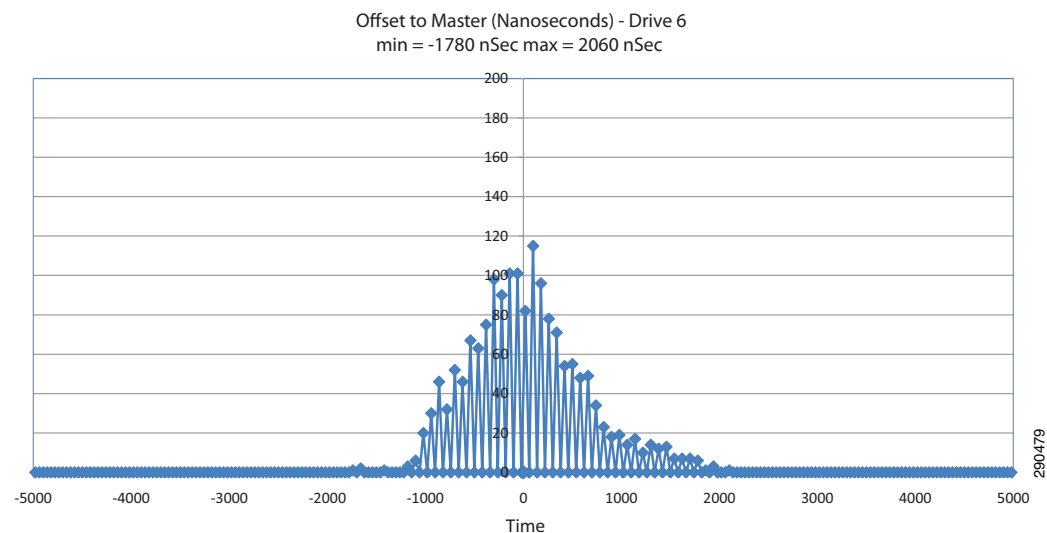


Figure 8-69 Star Architecture Phase Error Test 2.6—Test @ 50% Ixia Traffic Load

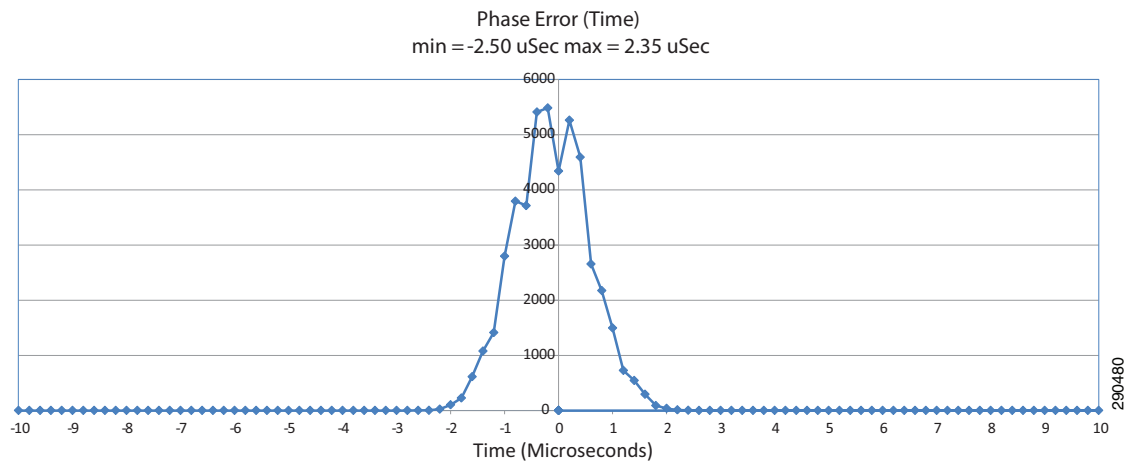


Figure 8-70 Star Architecture Position Error Test 2.6—Test @ 50% Ixia Traffic Load

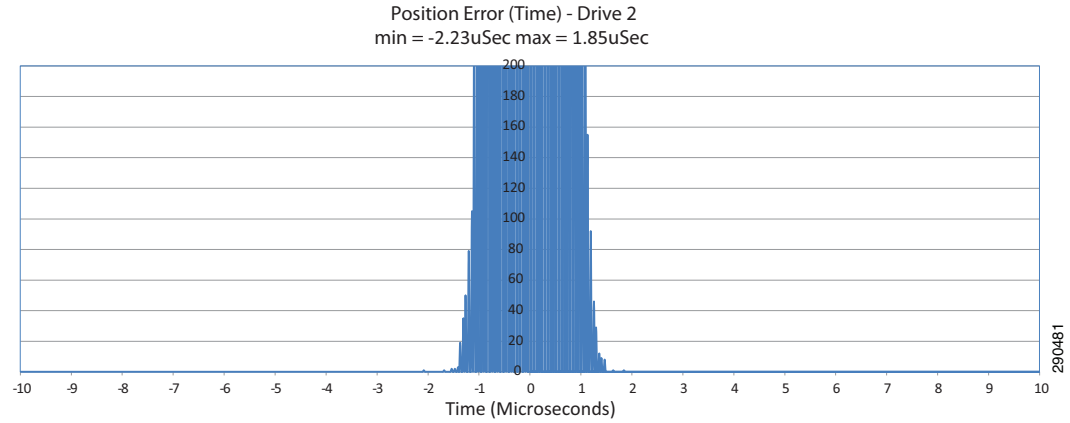
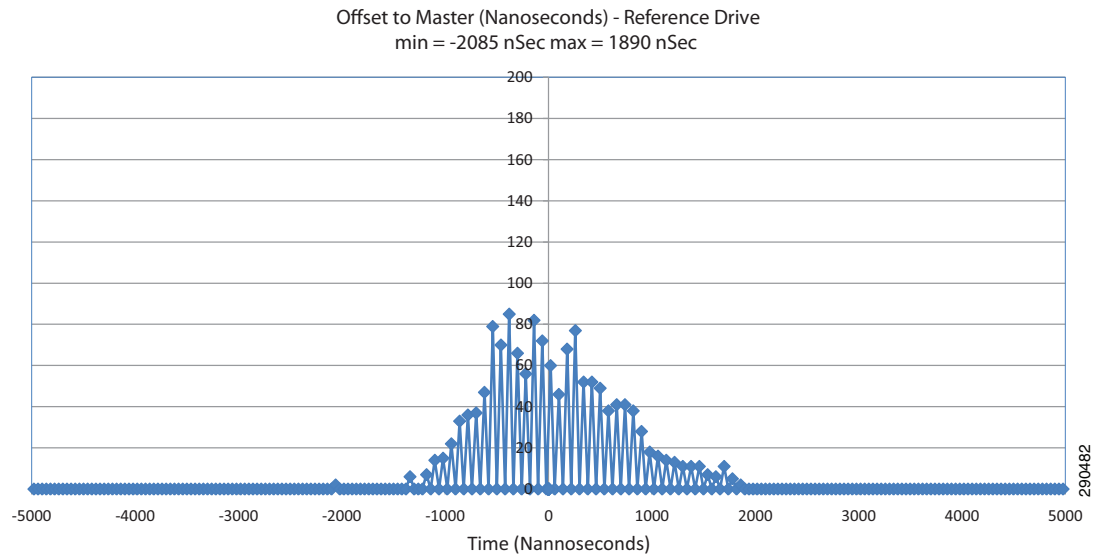


Figure 8-71 Star Architecture Offset to Master Test 2.6—Test @ 50% Ixia Traffic Load



Device-Level Ring (DLR) Architecture

Table 8-14 and Figure 8-72 through Figure 8-89 summarize the results of the DLR architecture test.

Table 8-14 DLR Architecture Test Results

Test 0.1: Test @ Nominal Ixia Traffic Load	
Test Procedure	Step 1Download the RefArchCIPMotion.acd program in the L7controller. Step 2Toggle the Bit Start_Test to start the test. Step 3Collect 100,000 samples. Step 4Open the Excel spreadsheet DataHandling_V7_ArchTest_A8_Linear_BaseTraffic.xls Step 5Click on Connect All Data Sheets to RSLinx Top (Logix Controller) button. Step 6Click on Read All Data from Logix Controller button. Step 7Wait for data to be read. Save the Excel file.
Results Summary	Phase Error: -3/2.01 μs Position Error: -2.17/1.68 μs Offset: -1.61/ 1.85 μs
Test 0.2: Test @ 10 Ixia Traffic Load	
Test Procedure	Step 1Download the RefArchCIPMotion.acd program in the L7controller. Step 2Toggle the Bit Start_Test to start the test. Step 3Collect 100,000 samples. Step 4Open the Excel spreadsheet DataHandling_V7_ArchTest_A8_Linear_10%Traffic.xls Step 5Click on Connect All Data Sheets to RSLinx Top (Logix Controller) button. Step 6Click on Read All Data from Logix Controller button. Step 7Wait for data to be read. Save the Excel file.
Results Summary	Phase Error: -2.84/2.17 μs Position Error: -2.28/1.96 μs Offset: -1.75/ 1.97 μs
Test 0.3: Test @ 20 Ixia Traffic Load	

Table 8-14 DLR Architecture Test Results (continued)

Test Procedure	Step 1Download the RefArchCIPMotion.acd program in the L7controller. Step 2Toggle the Bit Start_Test to start the test. Step 3Collect 100,000 samples. Step 4Open the Excel spreadsheet DataHandling_V7_ArchTest_A8_Linear_20%Traffic.xls Step 5Click on Connect All Data Sheets to RSLinx Top (Logix Controller) button. Step 6Click on Read All Data from Logix Controller button. Step 7Wait for data to be read. Save the Excel file.
Results Summary	Phase Error: -2.67/2.44 μs Position Error: -2.45/2.12 μs Offset: -1.98/ 1.84 μs
Test 0.4: Test @ 30% Ixia Traffic Load	
Test Procedure	Step 1Download the RefArchCIPMotion.acd program in the L7controller. Step 2Toggle the Bit Start_Test to start the test. Step 3Collect 100,000 samples. Step 4Open the Excel spreadsheet DataHandling_V7_ArchTest_A8_Linear_30%Traffic.xls Step 5Click on Connect All Data Sheets to RSLinx Top (Logix Controller) button. Step 6Click on Read All Data from Logix Controller button. Step 7Wait for data to be read. Save the Excel file.
Results Summary	Phase Error: -3/2.4 μs Position Error: -2.17/1.96 μs Offset: -1.7/1.99 μs
Test 0.5: Test @ 40% Ixia Traffic Load	
Test Procedure	Step 1Download the RefArchCIPMotion.acd program in the L7controller. Step 2Toggle the Bit Start_Test to start the test. Step 3Collect 100,000 samples. Step 4Open the Excel spreadsheet DataHandling_V7_ArchTest_A8_Linear_40%Traffic.xls Step 5Click on Connect All Data Sheets to RSLinx Top (Logix Controller) button. Step 6Click on Read All Data from Logix Controller button. Step 7Wait for data to be read. Save the Excel file.

Table 8-14 DLR Architecture Test Results (continued)

Results Summary	Phase Error: -2.78/2.27 μ s Position Error: -2.07/2.12 μ s Offset: -1.57/2.04 μ s
Test 0.6: Test @ 50% Ixia Traffic Load	
Test Procedure	Step 1Download the RefArchCIPMotion.acd program in the L7controller. Step 2Toggle the Bit Start_Test to start the test. Step 3Collect 100,000 samples. Step 4Open the Excel spreadsheet DataHandling_V7_ArchTest_A8_Linear_50%Traffic.xls Step 5Click on Connect All Data Sheets to RSLinx Top (Logix Controller) button. Step 6Click on Read All Data from Logix Controller button. Step 7Wait for data to be read. Save the Excel file.
Results Summary	Phase Error: -2.56/2.38 μ s Position Error: -2.07/1.74 μ s Offset: -1.53/2.09 μ s

Figure 8-72 DLR Architecture Phase Error Test 0.1—Test @ Nominal Ixia Traffic Load

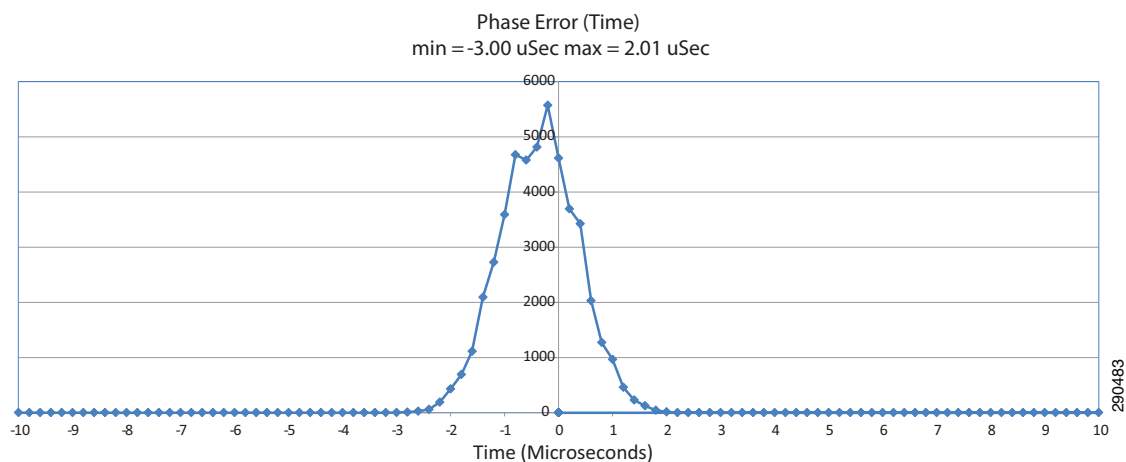


Figure 8-73 DLR Architecture Position Error Test 0.1—Test @ Nominal Ixia Traffic Load

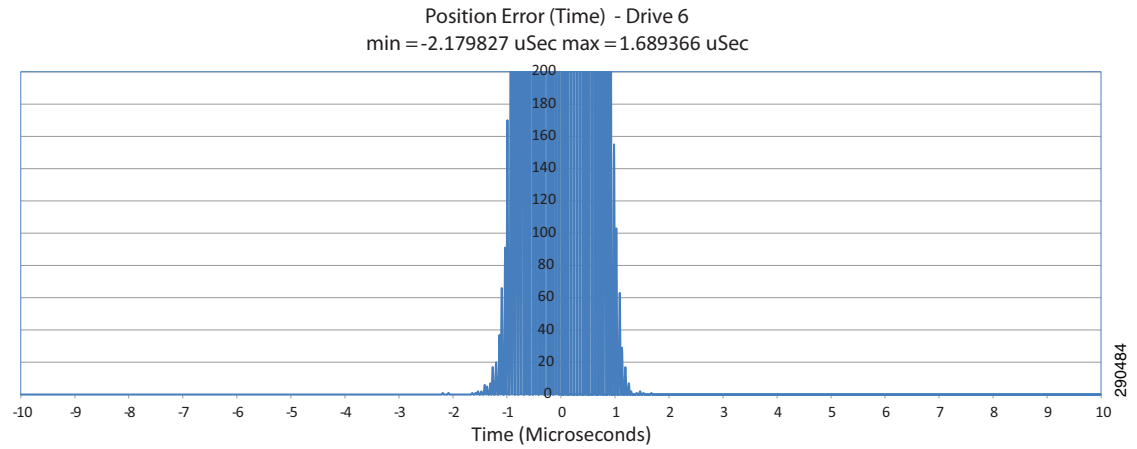


Figure 8-74 DLR Architecture Offset to Master Test 0.1—Test @ Nominal Ixia Traffic Load

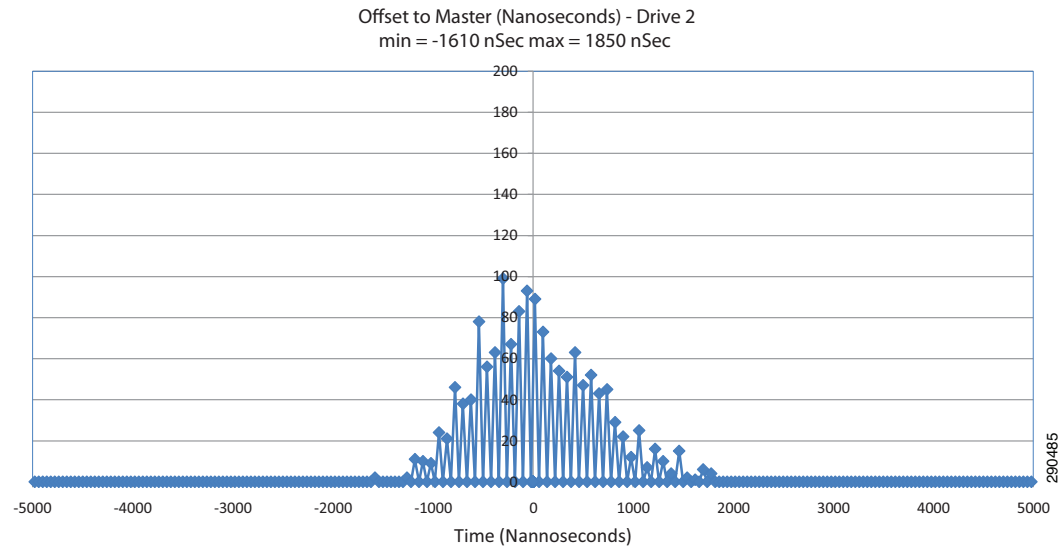


Figure 8-75 DLR Architecture Phase Error Test 0.2—Test @ 10% Ixia Traffic Load

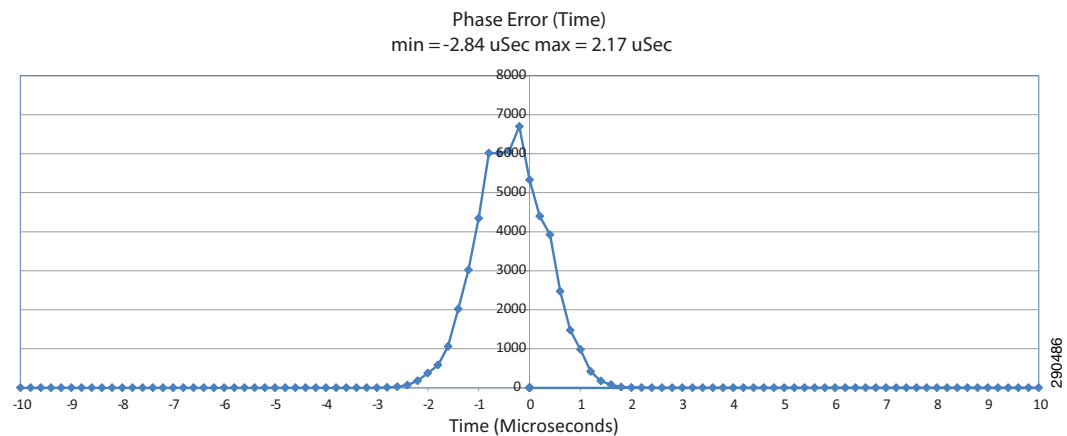


Figure 8-76 DLR Architecture Position Error Test 0.2—Test @ 10% Ixia Traffic Load

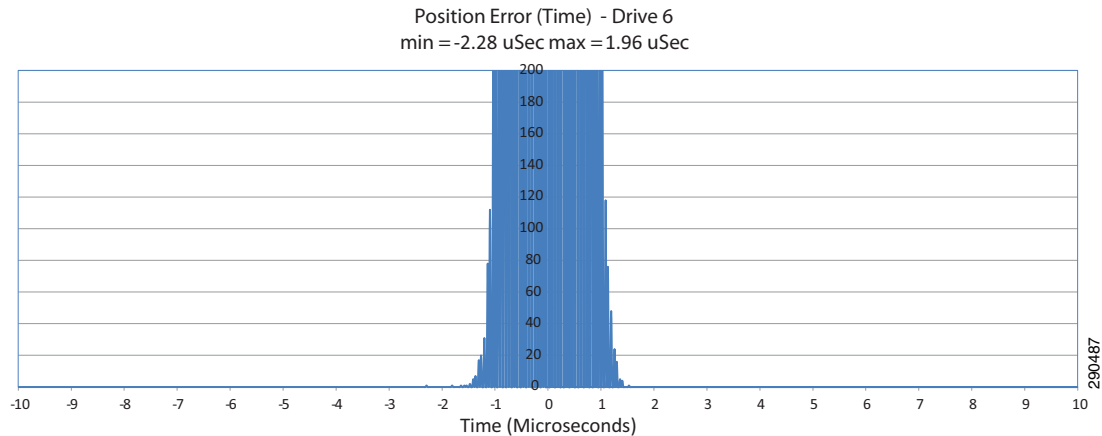


Figure 8-77 DLR Architecture Offset to Master Test 0.2—Test @ 10% Ixia Traffic Load

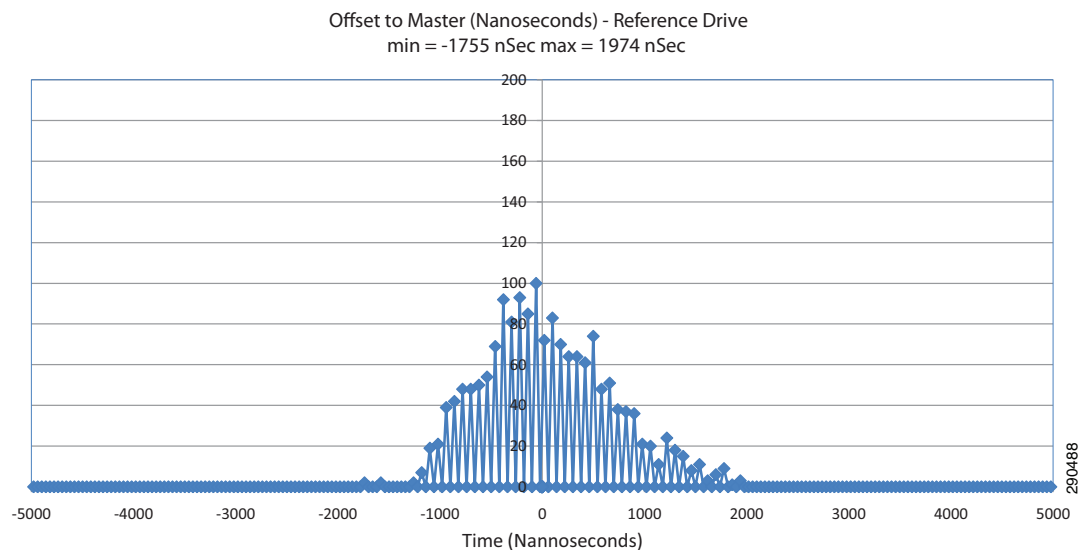


Figure 8-78 DLR Architecture Phase Error Test 0.3—Test @ 20% IX.IA Traffic Load

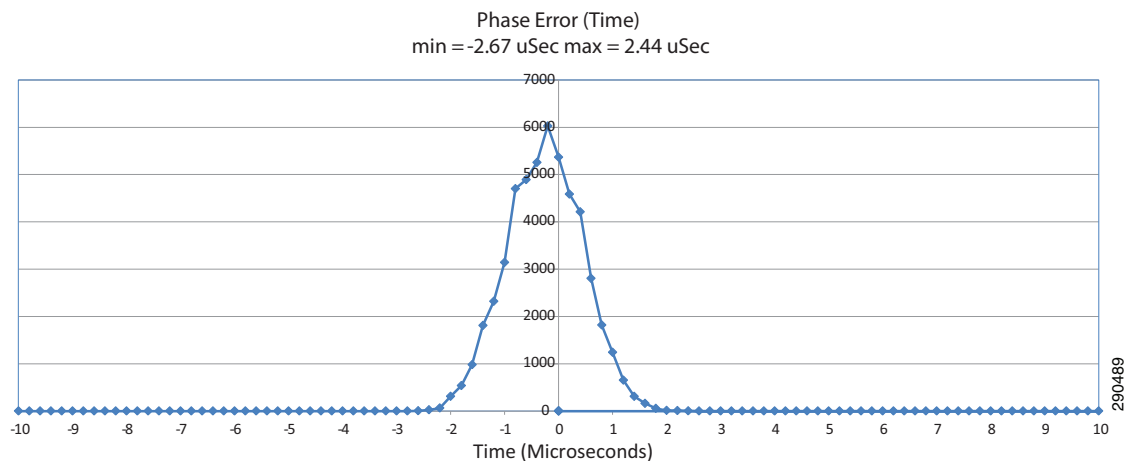


Figure 8-79 DLR Architecture Position Error Test 0.3—Test @ 20% Ixia Traffic Load

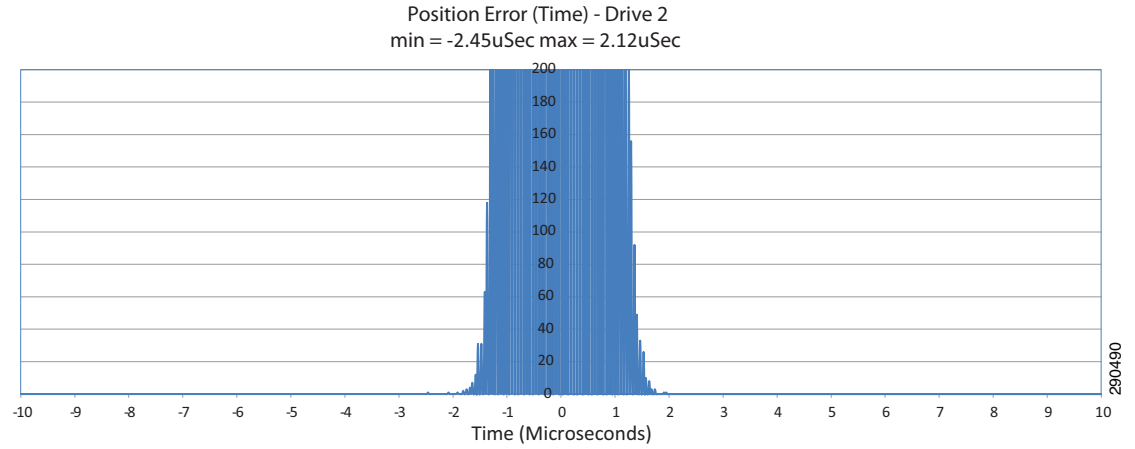


Figure 8-80 DLR Architecture Offset to Master Test 0.3—Test @ 20% Ixia Traffic Load

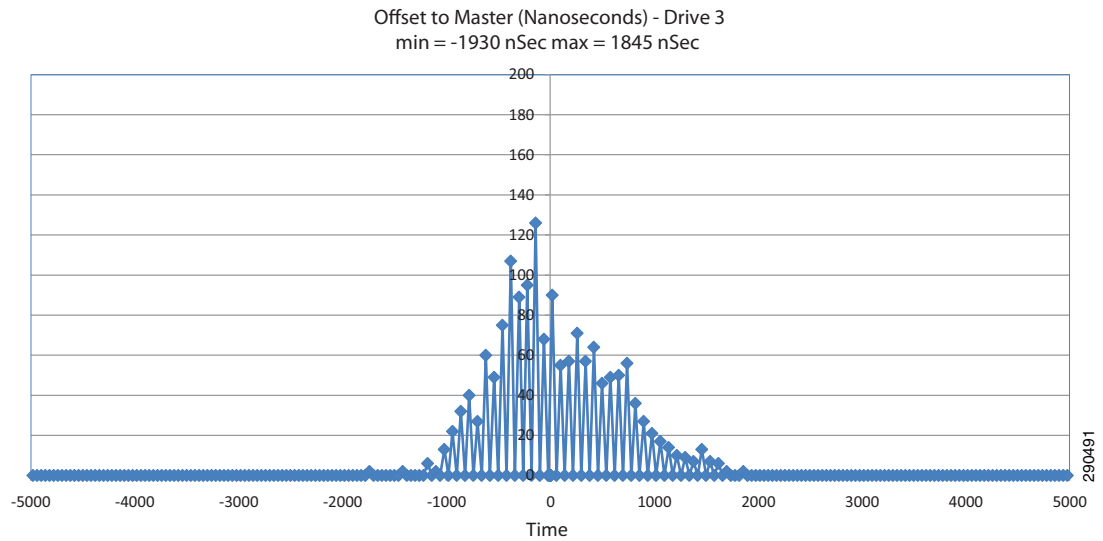


Figure 8-81 DLR Architecture Phase Error Test 0.4—Test @ 30% Ixia Traffic Load

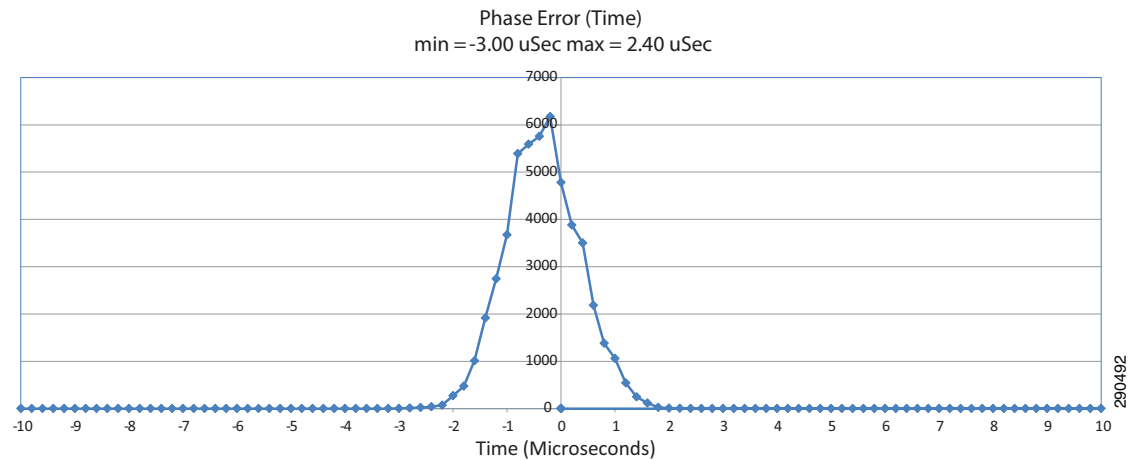


Figure 8-82 DLR Architecture Position Error Test 0.4—Test @ 30% Ixia Traffic Load

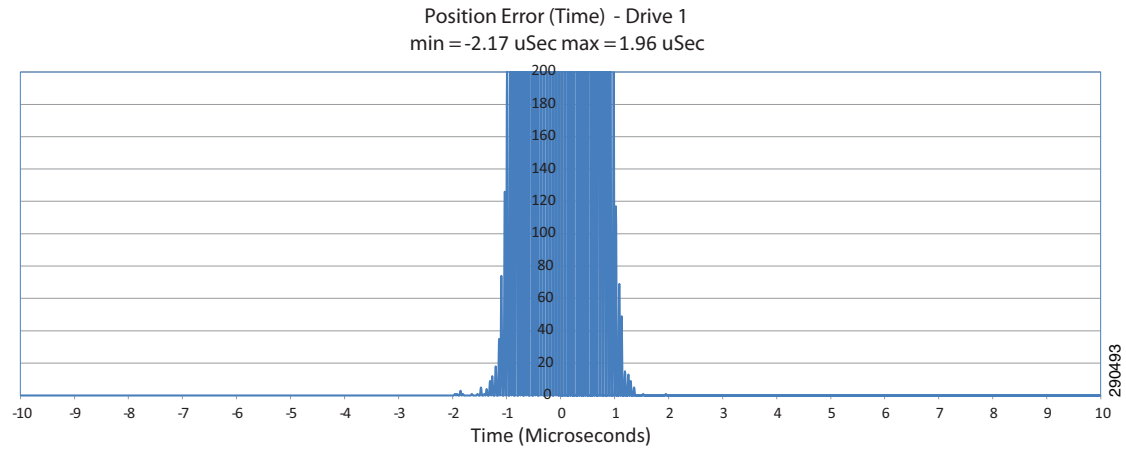


Figure 8-83 DLR Architecture Offset to Master Test 0.4—Test @ 30% Ixia Traffic Load

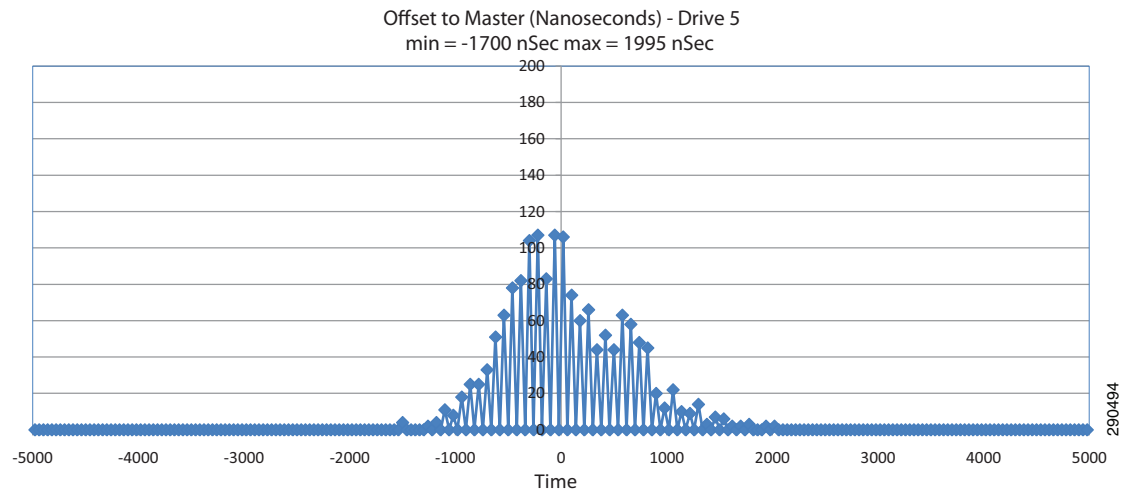


Figure 8-84 DLR Architecture Phase Error Test 0.5—Test @ 40% Ixia Traffic Load

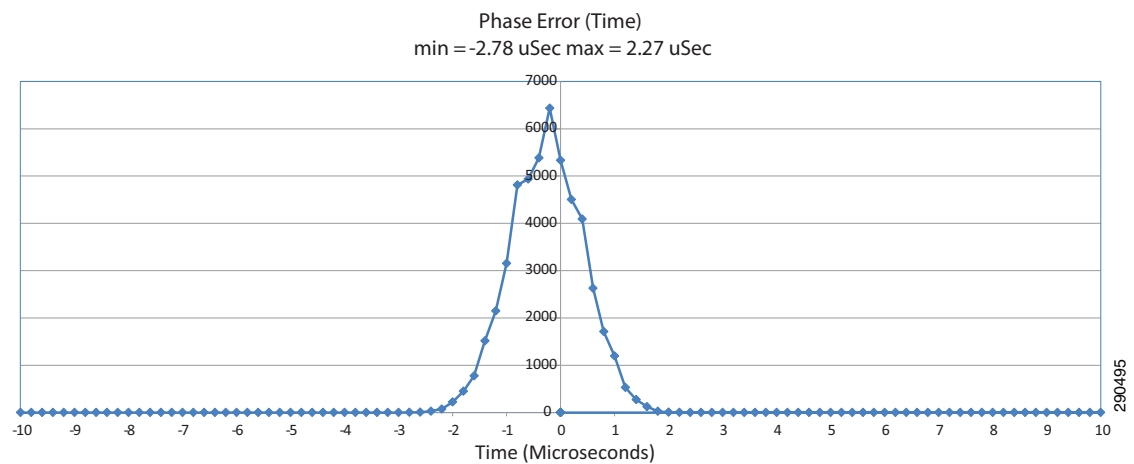


Figure 8-85 DLR Architecture Position Error Test 0.5—Test @ 40% Ixia Traffic Load

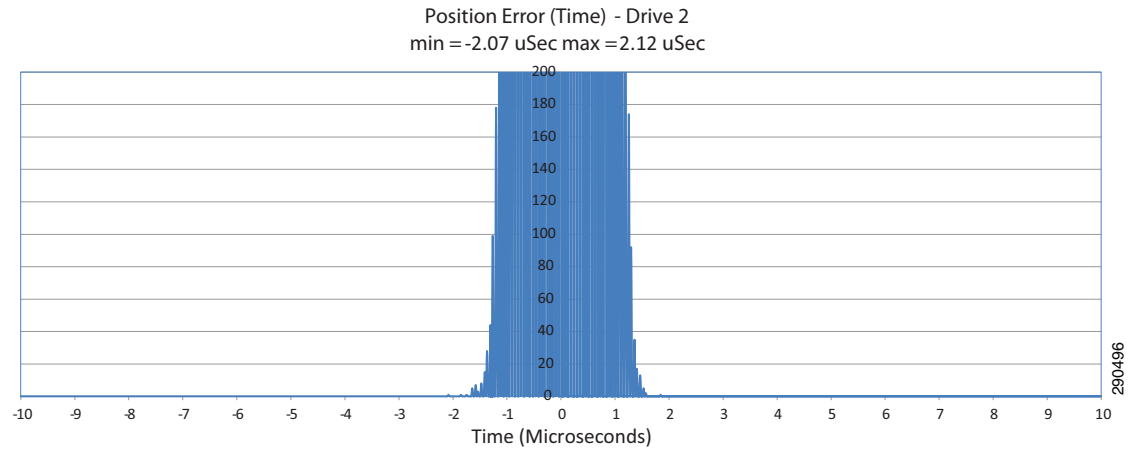


Figure 8-86 DLR Architecture Offset to Master Test 0.5—Test @ 40% Ixia Traffic Load

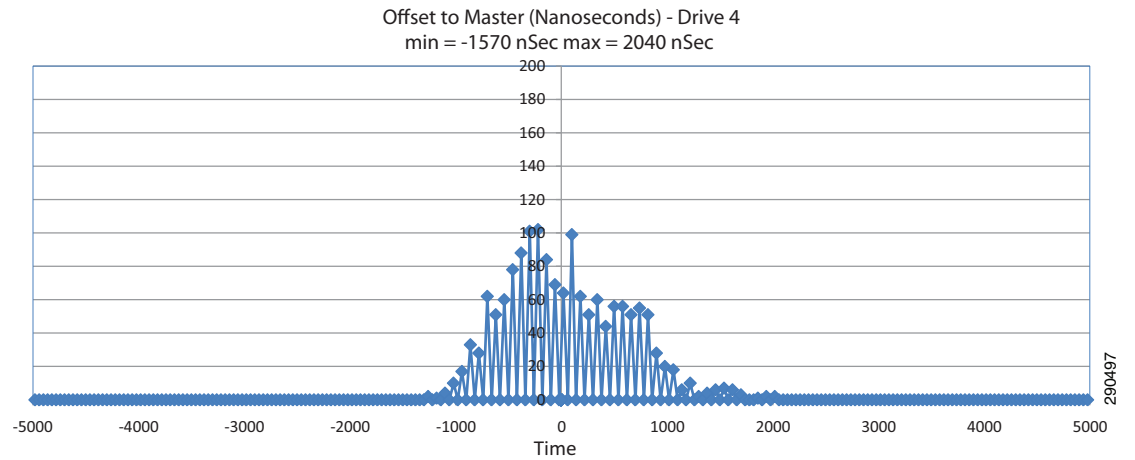


Figure 8-87 DLR Architecture Phase Error Test 0.6—Test @ 50% Ixia Traffic Load

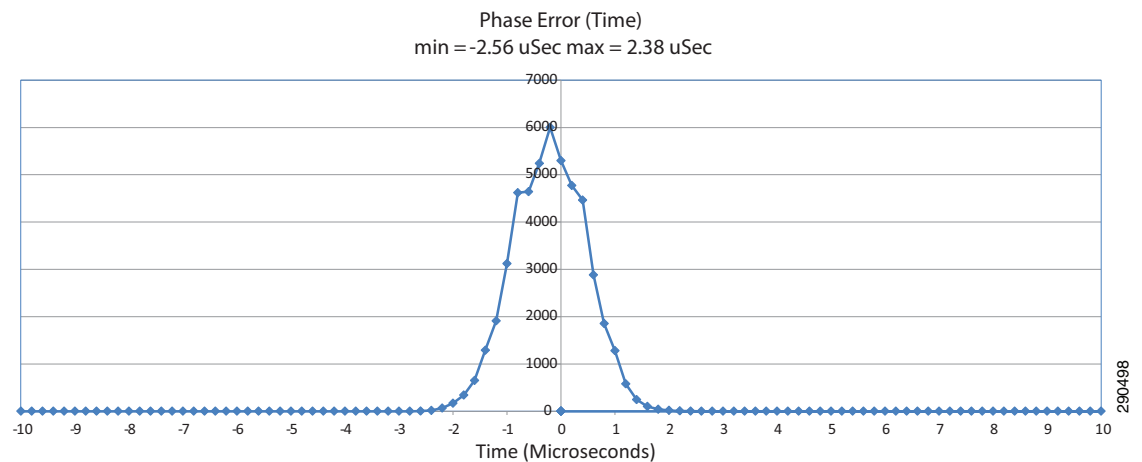


Figure 8-88 DLR Architecture Position Error Test 0.6—Test @ 50% Ixia Traffic Load

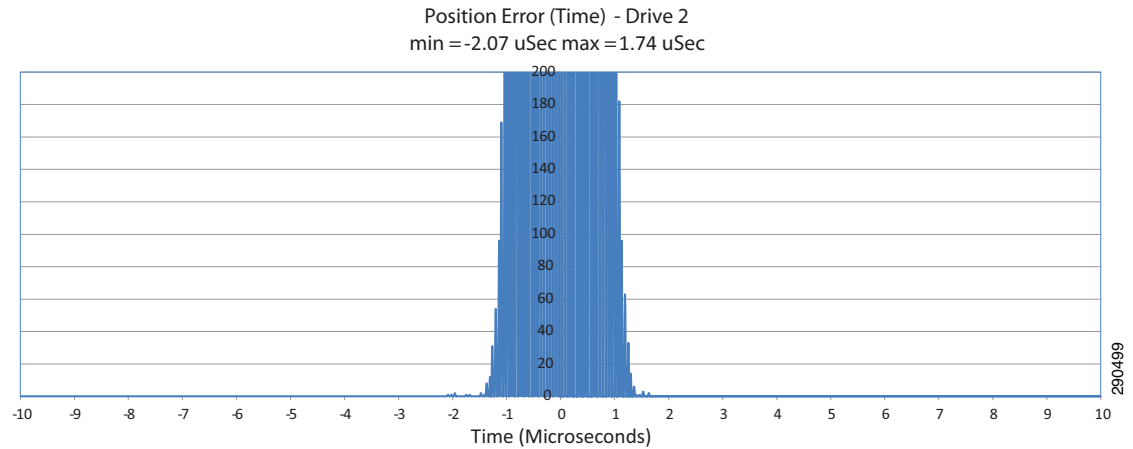


Figure 8-89 DLR Architecture Offset to Master Test 0.6—Test @ 50% Ixia Traffic Load

