CIP Motion System Performance Considerations

Mark Chaffee
Senior Principal Engineer
Rockwell Automation

Presented at the ODVA
2011 ODVA Industry Conference & 14th Annual Meeting
March 1-3, 2011
Phoenix, Arizona

Abstract:

This paper presents information helpful to anyone interested in evaluating or optimizing performance of an EtherNet/IP control system with CIP Motion drives. Two key principles that govern motion control performance are discussed, tight time synchronization and timely data exchange. To characterize data exchange performance, six key CIP Motion utilization metrics are defined based on the CIP Motion timing model, controller data processing performance, and Ethernet infrastructure performance, any of which could limit the overall performance of the system. We examine the impact various system components have on each of these metrics. These components include the motion controller, communications controller, Ethernet switches and cabling. The paper then highlights the important role End-to-End QoS plays in securing deterministic EtherNet/IP data delivery on an open network carrying non-time critical traffic. Good time synchronization performance, as measured by the Jitter metric, can be achieved through use of IEEE-1588 Boundary Clocks and Transparent Clocks, and further enhanced when used in conjunction with PTP Clock Filtering.

Representative EtherNet/IP network topologies are presented and performance considerations of these topologies are discussed. Network topologies can be partitioned into time critical device segments and non-time critical device segments. The role managed/unmanaged/embedded switches, QoS, and IEEE-1588 Boundary/Transparent Clocks have in these network topologies is also discussed with the end goal of maximizing performance while minimizing Ethernet infrastructure cost.

When equipped with contemporary system components, the maximum number of drives that can be supported for a given update period can be calculated for the system. Using multi-cycle CIP Motion timing models to relax timing constraints and full-duplex network data processing, motion control data exchange performance can reach 80-axes/msec over standard Fast Ethernet. When finally faced with the capacity limit of the 100 mbps Ethernet media, the paper takes a forward look to push the performance envelope using Gigabit Ethernet and Distributed Motion Control architecture.

Keywords:
CIP Motion, CIP Sync, End-to-End QoS, PTP Clock Filtering, Distributed Motion
Introduction:

Much of the attention in the industrial control marketplace concerning motion network performance has been focused on the protocol’s capacity on the Ethernet network. From an overall control system standpoint, claims that a motion control network protocol can support 100 axes per millisecond over the network media, are of little practical value. The Ethernet media is only one piece of the overall system performance puzzle and, most likely, one of the least limiting elements. This paper takes a holistic approach to characterize CIP Motion Performance, a system approach that has practical value and provides insight to a variety of performance optimization opportunities.

To begin a discussion of CIP Motion System Performance, we must first define the components of a typical EtherNet/IP Control System.

EtherNet/IP Control System Architecture

The system diagram below illustrates three general classifications of CIP data: Motion, I/O, and Messaging. System performance requirements dictate that this data be identified as High, Medium, and Low priority, respectively. Should system components be subjected to a mix of Motion, I/O, and Messaging data, these components must prioritize data processing to maintain deterministic data delivery for time critical data. This prioritized data processing strategy, as it applies to each of the control system components, is referred to as End-to-End QoS (Quality of Service).

Figure 1 – EtherNet/IP Control System Architecture

CIP Motion System Components

Our focus will be on the components within the red ellipse that are specifically associated with the CIP Motion system. On the left we have an automation controller that supports CIP Motion compliant drives. The controller executes a high priority, periodic, CIP Motion Task that begins by receiving fresh CIP Motion data from each drive.
associated with the D2C (Drive to Controller) connection. The Task then performs motion planner calculations to
determine the command reference value for each drive axis. Finally, the task assembles the C2D (Controller to
Drive) connection data block, including the command reference, and initiates transfer of the data block to the
Network Interface Component (NIC) via some form of bus interface. The bus interface mechanism could be DMA,
PCI, USB, etc. The NIC runs an EtherNet/IP stack that encapsulates the C2D connection payload in a standard
Ethernet UDP packet and transmits the packet to the targeted CIP Motion drive. The NIC typically consists of a
CPU to run the stack, a MAC to manage the Ethernet media, and a PHY to provide the hardware interface. The CPU
to run the stack can be the same CPU that runs the controller tasks, or can be a separate communications processor.
The MAC can also be a separate device, or as is commonly the case, integrated into the same CPU that runs the
stack. In the industrial control industry, the logical component that handles the various control tasks and application
program execution is sometimes referred to as the Automation Controller and the logical component that handles the
communications stack, MAC, and PHY are referred to as the Communications Controller.

After the NIC transmits the C2D packet on the Ethernet wire, the packet passes through one or more Ethernet
switches on its way to the targeted CIP Motion drive. These switches can be stand-alone, managed or unmanaged,
switches or they can be 3-port switches integrated into the drive product. Embedding 3-port switches in the drive is
particularly attractive in that it eliminates the cost of a separate switch and reduces cabling cost by allowing line or
ring network topology.

The packet is received by the drive’s NIC, which removes the Ethernet Encapsulation and delivers the CIP Motion
C2D payload to the drive’s application layer for processing. CIP Motion data processing is typically performed by
an interrupt driven CIP Connection Task. The CIP Connection Task processes the C2D connection data, updates
various axis object attributes, and establishes a new command reference target for the drive core to control the
motion of the associated motor and load.

Shortly before the start of the next update cycle, the CIP Connection Task assembles the latest data from the axis
object into a D2C connection data block and initiates transfer of the data block to the NIC via the bus interface. The
NIC runs an Ethernet/IP stack that encapsulates the D2C connection payload in a standard Ethernet UDP packet and
transmits the packet to the targeted controller. The NIC component functions on the drive side of the Ethernet
interface are the same as those on the controller side, but the communication demands on the controller side far
exceed those on the drive side. For this reason it is a common cost reduction strategy for the drive CPU to include an
integrated MAC and run the EtherNet/IP stack.

Following the same path as the preceding C2D packet, the D2C packet passes back through the Ethernet switches on
its way to the targeted CIP Motion controller. The controller NIC receives the D2C packet, strips off the Ethernet
encapsulation and initiates transfer of the D2C payload to the controller’s memory via the bus interface, thus
completing the connection cycle.

Each of these components plays an important role in the overall performance of the CIP Motion system.

CIP Motion Timing Model

The motion control industry has generally adopted the “axis per msec” metric as an indicator of motion control
system performance. This metric represents the number of axes the system can control for a given connection update
period, or cycle time. Generally, the performance of a given motion control system improves as the update rate
increases. For a motion control system, the axis per msec performance level is determined by the constraints of the
timing model.

The CIP Motion 1-Cycle Timing Model shown in Figure 2 below defines the exchange of data between the motion
controller and the drive device.
This timing model breaks the update cycle (sometimes referred to as the Coarse Update Period) into three equal parts, the input interval, the planner interval, and the output interval. The timing is quite simple:

1. Input data must arrive at the controller before the end of input interval, or 1/3 update period.
2. Planner generated output should be transmitted before the end of the planner interval, or 1/3 update period.
3. Output data must arrive at the drive before the end of the output interval, or 1/3 update period.

**System Utilization Metrics**

These above three timing constraints are the basis for three Cycle Utilization metrics:

1. Input Cycle Utilization - % of CIP Motion update cycle allotted to D2C input packet transfer. This value should be less than 100% to insure D2C data delivered to controller before Motion Task starts.
2. Output Cycle Utilization - % of CIP Motion update cycle allotted to C2D input packet transfer. This value should be this less than 100% to insure C2D data delivered to drive before next cycle starts.
3. Motion Task Cycle Utilization - % of CIP Motion update cycle allotted to process input packets, run planner, and assemble output packets. This value should be less than 100% for timely C2D data delivery.

As more axes are added to the control system, or alternatively, as the connection update rate of the system is increased, the above Cycle Utilization metric values also increase. At the point where the values of any one of these three metrics reaches 100%, the CIP Motion system has reached its axis/msec performance limit.

There are three more Utilization metrics that involve the capacity of key components of the control system.
4. Automation Controller Utilization - % of Automation Control processor capacity used to execute Motion Task. Generally this value should be less than 50% to allow for Application Program execution, I/O and Message data processing, and other controller tasks. The Automation Controller Utilization limit may or may not be enforced.

5. Communications Controller Utilization - % of Communications Control (or NIC) processor capacity used to process motion packets. This value should be less than 100%, perhaps less than 50% if significant non-motion Class 1 connection data processing is required. The Communications Controller Utilization limit is often enforced for Class 1 connections by the Transmission Specification (T-Spec).

6. Ethernet Media Utilization - % of Ethernet media capacity consumed by CIP Motion packet traffic. This value should be less than 100%, perhaps less than 50% if significant non-motion Class 1 connection data transfer is required over the same network segment.

For a representative 50-axis machine application, these six Utilization metrics can be plotted as a function of Coarse Update Period (update cycle time) as shown in Figure 3 below. Such plots can be used to determine the performance limits of a CIP Motion control system. In this case, the performance bottleneck of this 50-axis system happens to be the three Cycle Utilization metrics that reach 100% at a Coarse Update Period of 2 msec.
If the timing constraints associated with the 1-Cycle Timing Model were relaxed, the performance of this CIP Motion system could be significantly improved. This is the motivation behind CIP Motion’s provision for multi-cycle timing models.

**Multi-Cycle Timing Models**

CIP Motion can also support a 2-Cycle Timing Model, shown in Figure 4 below, that nearly doubles the amount of time available for input and output data transfer and motion planner calculations. In the 2-Cycle Timing Model the CIP Motion input data is transferred during the first half of the cycle, processed by motion task, with the associated output data transmitted during the later half of the next cycle. The Motion Task in the controller, that is responsible for running the motion planner, is scheduled to start half way into the current cycle. It is interesting to note that timing models associated with other motion network protocols, e.g. SERCOS, are typically 2-cycle models.

The 2-Cycle Timing Model breaks the update cycle, or Coarse Update Period, into two equal parts, the input interval and the output interval. The timing constraints are:

1. Input data must arrive at the controller before the end of input interval, or 1/2 update period.
2. Planner generated output computed before the end of the current cycle, or 1/2 update period.
3. Output data must arrive at the drive before the end of the output interval, or 1/2 update period.

As a direct result of the relaxed timing constraints associated with the 2-Cycle Timing Model, there is a substantial increase in performance as shown in Figure 5 below. Note that the three Cycle Utilization curves now reach 100% at a Coarse Update Period of 1 msec, as does the Communications Controller Utilization. But note that the Automation Controller Utilization has reached 67% at 1 msec, which is higher than the recommended 50% upper limit.
Following the Automation Controller Utilization curve, we see that it reaches 50% at a Coarse Update Period of 1.5 msec. So by using the 2-cycle timing model we were able to boost the performance of the CIP Motion system from 25-axes/msec to 33-axes/msec before the next performance bottleneck was reached.

With the Automation Controller Utilization being the performance bottleneck, the obvious next step in improving CIP Motion system performance is to increase the processing power of the Automation Controller. This has the effect of reducing both the Automation Controller Utilization and the Motion Task Cycle Utilization curves. Figure 6 shows the impact of doubling the processing power of the Automation Controller. With the 2-cycle timing model and this higher powered Automation Controller, we have increased the performance of the CIP Motion system from 33-axes/msec to 50-axes/msec.

Another, not so obvious, solution to the Automation Controller bottleneck is to lift the 50% limit on Automation Controller Utilization. This can be done by executing Motion Task and the Application Program on separate cores of a multi-core processor. Automation Controller Utilization would then be free to use the full capacity of the dedicated CPU core without impacting Application Program scan time to achieve the 50 axes/msec performance level.
The System Utilization curves of Figure 6 suggest by further lifting cycle timing constraints and boosting Communications Controller performance, we could squeeze even more performance out of the CIP Motion system. This can be done by supporting a 3-Cycle Timing Model as shown in Figure 7. This timing model requires both input traffic and output traffic to traverse the Full-Duplex Ethernet infrastructure simultaneously and also implies parallel input and output packet data processing.

The 3-Cycle Timing Model uses the entire update cycle, or Coarse Update Period, for both the input and output data transmission. The timing constraints are:

1. Input data must arrive at the controller before the end of the 1st cycle, or 1 full update period.
2. Planner generated output computed before the end of the 2nd cycle, or 1 full update period.
3. Output data must arrive at the drive before the end of the 3rd cycle, or 1 full update period.
To reap the relaxed timing constraints of this model, the Communications Controller must support parallel processing of input and output traffic. Full Duplex data processing can be accomplished using separate cores of a multi-core processor, effectively doubling the overall data processing capacity of the Communications Controller component.

![Figure 7 – CIP Motion 3-Cycle Timing Model](image)

With the relaxed timing constraints associated with the 3-Cycle Timing Model and Full Duplex data processing, CIP Motion System Performance has increased from 50-axes/msec to 80-axes/msec as shown in Figure 8 below. But note that Ethernet Media Utilization has also increased to nearly 90% of capacity.
Utilization vs Coarse Update Period

- Automation Controller Utilization
- Comm Controller Utilization
- Input Cycle Utilization
- Output Cycle Utilization
- Motion Task Cycle Utilization
- Ethernet Media Utilization

Figure 8 – CIP Motion System Utilization Plot – 80-Axes, 3-Cycle Timing, 2x AC Boost, Full-Duplex

With all our performance optimizations in place, we have finally reached the capacity of 100 Megabit/sec Fast Ethernet! Where do we go from here?

**Gigabit Ethernet**

There is no question that Gigabit Ethernet will relieve any future Ethernet Media Utilization constraints we may have, but will it have significant impact on CIP Motion System Performance? Figure 9 shows performance curves for this same CIP Motion System equipped with Gigabit Ethernet media. As can be seen from the curves, we still have constraints imposed by data processing delays both in the Automation Controller and the Communications Controller. In order to take advantage of Gigabit Ethernet, we’ll have to continue to push the data processing power of both the Automation Controller and Communications Controller CPUs.
Figure 9 – CIP Motion System Utilization Plot – 80-Axes, 3-Cycle Timing, 2x AC Boost, Full-Duplex, Gigabit

**Distributed Control**

Perhaps a better way to advance CIP Motion System Performance is to distribute the Motion Planner function to the individual CIP Motion devices on the network. This would reduce the computational burden on the Automation Controller. Furthermore, with the Motion Planner distributed to individual CIP Motion devices, there is no need to run the CIP Motion connections at high update rates. So instead of running this system using a 1 msec update period we could provide equivalent performance using, say, a 10 msec update period. As can be seen in the above plots, this would greatly reduce the connection data processing burden for both the Automation Controller and the Communications Controller.

**End-to-End QoS**

The above analysis assumes a private Ethernet network dedicated to the transfer of CIP Motion connection data. This is analogous to traditional motion protocols like SERCOS, PROFINET, and EtherCAT that are based on hard synchronization, where timely data delivery is guaranteed by scheduling algorithms and special hardware to preclude non-motion traffic. Non-motion traffic in these systems is channeled into a special time slice reserved for
standard Ethernet packet traffic. By contrast, CIP Motion allows both motion and non-motion traffic to flow freely through the network infrastructure to serve Drives, I/O, and HMI devices, PC Programming Stations, and even connectivity to IT through a standard Router. What makes deterministic and timely motion data delivery possible in the presence of non-motion traffic are standard QoS protocols for both Ethernet and CIP. The application of these Ethernet QoS and CIP QoS protocols throughout the entire connection path is referred to as End-to-End QoS.

There are two standard Ethernet QoS protocols available. The IEEE-802.1D Tagged Frame packet protocol uses a special three-bit QoS field in the extended Ethernet Frame header to determine the priority of the packet. This Layer 2 (Data Link Layer) protocol adds an additional field to the standard IEEE 802.3 format. As a result, any network components that do not support the Tagged Frame protocol would have to reject this packet as a non-standard format, thus negatively impacting interoperability. For this reason, ODVA strongly recommends the DSCP (Differentiated Service Code Point) protocol as its preferred EtherNet/IP QoS solution. DSCP utilizes the existing Type of Service (ToS) field in the IP header to determine the priority of the packet. The CIP Motion Drive Device Profile specification echoes the ODVA recommendation stating that all CIP Motion devices shall implement DSCP marking for CIP Motion packets.

In order to get value from DSCP marking, all Ethernet components in the motion connection path must use the DSCP marking to prioritize queue data processing. These components include the Ethernet switches, the NIC receive stack, and the CIP Motion device receive stack. But what about the transmit stacks and the internal bus interface to the NIC that are handling connection data in the CIP Application Layer? What puts the CIP Motion connection data on the fast track through these components versus other non-motion data? Clearly, Ethernet QoS must extend into these components to provide an End-to-End solution. ODVA recently addressed this issue in Chapter 3 of Volume 2, the EtherNet/IP Adaptation of CIP specification by mapping DSCP codes to CIP Priority levels as shown in Table 1. CIP Motion packets are assigned a CIP Priority of Urgent. When a CIP component is faced with the decision of processing motion data versus non-motion data in the processing queue, the Urgent priority level trumps all other priority levels, so the motion data is immediately processed.

<table>
<thead>
<tr>
<th>Traffic Type</th>
<th>CIP Priority</th>
<th>DSCP</th>
<th>CIP Traffic Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>PTP event (IEEE 1588)</td>
<td>n/a</td>
<td>59 (‘111011’)</td>
<td>PTP event messages, used by CIP Sync</td>
</tr>
<tr>
<td>CIP Class 0 / 1</td>
<td>Urgent (3)</td>
<td>55 (‘110111’)</td>
<td>CIP Motion</td>
</tr>
<tr>
<td>Scheduled (2)</td>
<td></td>
<td>47 (‘101111’)</td>
<td>Safety I/O, I/O</td>
</tr>
<tr>
<td>High (1)</td>
<td></td>
<td>43 (‘101011’)</td>
<td>I/O</td>
</tr>
<tr>
<td>Low (0)</td>
<td></td>
<td>39 (‘100111’)</td>
<td>Open</td>
</tr>
<tr>
<td>CIP UCMM CIP Class 3</td>
<td>All</td>
<td>35 (‘100011’)</td>
<td>CIP Messaging</td>
</tr>
</tbody>
</table>

Table 1 – QoS Priority Mapping

Deterministic Data Delivery

With support for End-to-End QoS, it is possible to determine the maximum delays that CIP Motion data might incur as it passes through the connection components. For example, the maximum delay that a CIP Motion packet may experience in passing through a switch supporting QoS is the time for the switch to processes a non-motion packet, which is approximately the transmission time of the packet. Suppose the maximum non-motion packet size is limited to 500-bytes, a good assumption for a network consisting of CIP connection traffic. Based on a 100 Mbps Ethernet data rate, the additional motion packet data transfer delay is approximately 40 microseconds, worst case.
The effect an occasional 40 microsecond delay has on overall motion system performance is limited to a small reduction in the maximum axis per msec that the system can support, or approximately a 1 to 2 axes per millisecond impact on motion performance. The worst case scenario is a mix of 1500-byte non-motion packets. This results in a maximum incremental delay of 120 microsecond on motion packet transfer time, or approximately 5 to 6 axes per millisecond impact on motion performance.

Note that because the CIP Motion information is time stamped using CIP Sync technology, this occasional incremental delay for servicing a non-motion packet does not adversely impact the quality of real-time control of the axes. Command position values are applied by the drive based on the associated time stamp, independent of the exact time of delivery. As long the quality of CIP Sync clock synchronization between the CIP Motion device and the controller is good, the quality of real-time control shall also be good. This is a good segue to our next topic on CIP Sync performance considerations.

**CIP Sync Performance**

The performance of time synchronization between the controller and CIP Motion devices is dependent on timely data delivery of the Time Sync messages across the Ethernet infrastructure. Any jitter in the propagation delay in delivering Time Sync messages from the PTP Grand Master to a CIP Motion controller or drive translates to Time Jitter that has a direct impact on motion profile quality. Thus, when a drive computes the local clock target time to apply a Command Position value based on its associated time stamp, a disturbance in PTP Time translates to a disturbance in the computed target time, which in turn produces a disturbance to the motion profile. While, switch-based QoS helps to mitigate this effect by bounding the delay impact of non-motion traffic, more can and should be done.

**Boundary Clocks & Transparent Clocks**

To reduce jitter in Time Sync message propagation delay from the PTP Grand Master to CIP Motion controllers and devices, it is recommended that all intervening switches support either IEEE-1588 Boundary Clocks or Transparent Clocks. 1588 Boundary Clocks effectively replicate the Grand Master clock to eliminate delay disturbances that would normally be imposed on PTP packets as they pass through the switch. 1588 Transparent Clocks take a different approach to the problem. Rather than replicating the Grand Master Clock in the switch, the Transparent Clock function simply adjusts the PTP time stamp in the Time Sync message by the amount of delay added by the switch prior to transmission. Both techniques are very effective in eliminating the impact of switch delay on the quality of time synchronization between CIP Motion controllers and drives, resulting in very low Time Jitter.

**PTP Clock Filtering**

Another promising technique to improve the quality of time synchronization between CIP Motion controllers and drives is PTP Clock Filtering. This approach simply applies a low pass filter to the PTP Time signal to reduce Jitter. The basis for this is the fact that the local clocks in CIP Sync devices are highly inertial. The frequencies of these clocks change very slowly, primarily due to ambient temperature variation. PTP Time variations from one update to the next, typically 1 second apart, are almost entirely due to propagation delay noise. Thermal time constants surrounding the clock crystals are much longer than 1 second, more likely on the order of minutes, so a 10 to 20 second low-pass filter can greatly attenuate the noise component of the PTP Time signal while still preserving the low frequency correction component of the signal. The effectiveness of this technique has been demonstrated in motion control systems with unmanaged switches, i.e. low cost consumer switches with no built-in QoS, or 1588 Clock support.
Network Topology Considerations

CIP Motion devices can be connected to the controller using a wide variety of Ethernet network topologies; line, ring, star, or any combination thereof. The performance considerations outlined above provide valuable guidance in laying out the network architecture for optimal machine control.

Clearly, all switches between the controller and the drives should support QoS and IEEE-1588. If the number of switches between the controller and the drive is small, i.e. a star topology, IEEE-1588 support can be either via Boundary Clock or Transparent Clock. If the number of switches between the controller and the drive is large, i.e. a line or ring topology, IEEE-1588 support should be via Transparent Clock. Having more than 5 cascaded Boundary Clocks is not recommended, while more than 60 cascaded Transparent Clocks in a line topology have been successfully demonstrated.

When laying out the network topology it is good practice to separate non-time critical traffic from time critical traffic into two separate network segments. Switches servicing the non-time critical traffic segment do not require QoS or IEEE-1588 support since time synchronization does not need to be precise, nor does data delivery need to be deterministic. Conventional lower-cost unmanaged switches can therefore be used for this part of the network. By contrast, switches servicing the time-critical traffic segment do need to provide support for QoS and IEEE-1588. A convenient and cost-effective way to provide this support is to utilize the embedded 3-port switch technology that is part of ODVA’s CIP Sync toolkit. These high-performance 3-port switches are available as standalone devices or as part of an FPGA built into the CIP Motion device and controller. Using these 3-port switch devices, the time-critical side of the network can be connected in its entirety without the help of an expensive managed switch. Figure 10 is an example of such a network topology. The 3-port switch, in this case, is embedded in the Controller’s communications module equipped with two external ports. One external port of this switch is connected to the Ethernet segment serving Time Critical components (Motion and I/O) on the right and the other external port is connected to the Ethernet segment serving Non-Time Critical traffic components (EOI, PC) on the left. The external switch on the left can in this case be a low cost unmanaged switch.

![Control network topology with separate segments for Time Critical (R) and Non-Time Critical traffic (L)](image)

Using 3-port switches allow for a variety of Line topologies such as those as shown in Figures 11A, 11B and 11C, and Device Level Ring topologies as shown in Figure 11D. From a motion performance standpoint, the best practice is to segregate Time Critical traffic from Non-Time Critical traffic. In this regard, topologies represented by Figures
11B, 11C and 11D have an edge on 11A since Class 3 traffic from the EOI device shown in 11A can theoretically delay downstream Time Critical packets by multiple Class 3 packet transmissions.

![Figure 11 - Line and Ring network topologies using embedded 3-port switch technology.](image)

**Conclusion**

In this paper we have demonstrated the importance of a holistic system view when analyzing motion control performance over Ethernet. Our analysis of CIP Motion system performance uncovered 6 different utilization metrics, any of which could become a motion performance bottleneck. Three of those metrics are related to timing constraints associated with the CIP Motion timing model. We saw how these constraints could be lifted by applying multi-cycle timing models. Two of the utilization metrics were related to the data processing power of the Automation Controller and the Communications Controller. Only after applying very fast, multi-core, CPU technology to the Automation Controller and Communications Controller components do we finally reach the capacity limit of 100-mbps Ethernet, as characterized by our sixth and final performance metric, Ethernet Media Utilization. Pressing beyond the performance limits imposed by 100-mbps Ethernet naturally leads us to consider Gigabit Ethernet or Distributed Control architecture.

We then examined the performance impact of adding non-motion traffic to the CIP Motion system and the role of End-to-End QoS plays in securing deterministic time-critical data delivery not only for CIP Motion but also for CIP Sync. Since the quality of motion control is tied directly to the quality of time synchronization between the controller and drive components achieved by IEEE-1588 based CIP Sync, it is important that switches associated with time critical traffic support either IEEE-1588 Boundary Clocks or Transparent Clocks. When CIP Motion system components support these standard QoS and IEEE-1588 protocols, the quality of motion control is substantially independent of the volume of non-motion traffic on the network components and, yet, this is still standard Ethernet!