A Distributed Ethernet Network of Advanced Pedestrian Signals

Submitted by

Dustin DeVoe
Electrical Engineering Student
Department of Electrical and Computer Engineering
University of Idaho
PO 441023
Moscow, ID 83843-1023

Sanjeev Giri
Electrical Engineering Student
Department of Electrical and Computer Engineering
University of Idaho
PO 441023
Moscow, ID 83843-1023

Richard W. Wall, P.E.
Professor
National Institute for Advanced Transportation Technology
Department of Electrical and Computer Engineering
PO 441023
University of Idaho
Moscow, ID 83843-1023
208.885.7226
rwall@uidaho.edu

Submitted for presentation at the 88th Annual Meeting of the Transportation Research Board, January 11th - 15th, 2009 and for publication in Transportation Research Record, the Journal of the Transportation Research Board.
Abstract: For over 60 years, traffic signals have used direct wire connections between the traffic controller cabinet and the signals and detectors dispersed throughout the intersection. This paper reports on a networked based approach for distributed control and sensing of traffic signal devices. The motivation of our research has been to improve safety and performance while reducing the cost of a signalized intersection installation. We present an Ethernet based network architecture that uses NTCIP for real-time signal control that is combined with the IEEE 1588 precision time protocol for robust operation of safety critical applications. Our investigation focuses on improving the accessibility and safety for pedestrians through the use of Smart Signals.

The Smart Signals paradigm is the basis of an enabling technology that permits complex functioning signals and detectors with self-test capability. The bi-directional communications provides the ability to monitor the operational status of signals and detectors that are currently unobservable by the automated traffic controls. The elements of network messaging that enable NTCIP to be used for real-time intersection signal control are presented. The concerns of signals outputs being generated outside the scope of observation for malfunction management units are addressed by implementing a deterministic network. The time performance of the system is evaluated for SNMP and STMP network messages. The results of test on the global network time synchronization are presented and show that inexpensive microprocessors can achieve stable long-term time division multiplexed operation with one hundred microsecond accuracy.
INTRODUCTION

The fundamental means for controlling traffic signal lights is and has been since its inception, over 60 years ago, to have dedicated wires for controlling signal lights and detecting vehicles and pedestrians in a binary fashion [1]. In 2006, we presented a networked architecture for controlling traffic signal lights based on the IEEE 1451 Family of Transducer Interface Standards that results in the ability to exchange more complex information than simple binary states of indication and detection [2,3]. The Plug and Play (IEEE 1451) standard was initially chosen because it supported the capabilities to potentially simplify intersection signal installations, operation, and upgrades. A path to integration was presented in 2007 that demonstrated how the IEEE 1451 based signals and detectors can be integrated with modern TS2 controllers [4]. Since the networked signals and detectors now contain innate intelligence, we refer to the system as “smart signals”.

In 1992, the National Electrical Manufacturers Association (NEMA) approved the TS2 standard which specified an open architecture. This provided more robust methods of fault tolerance and information distribution inside traffic controller cabinets. In this standard, the TS2 Port 1 serial data bus (SDLC) replaced direct point-to-point wiring methods that often cluttered cabinets with proprietary solutions. The SDLC represents a distributed control environment in which the traffic controller, load switches, detectors and malfunction management unit (MMU) share a common communication bus. Each component has particular message types defined in TS2 as frames that must propagate the network in a time regulated network [5].

The purpose of this paper is to describe how a pedestrian signals and buttons can be controlled in a fault tolerant distributed Ethernet network using standardized protocols. National Transportation Communications for Intelligent Transportation System (ITS) Protocol (NTCIP) defines objects which can provide control and state information for efficient operations of a compliant traffic system [6]. We conducted research which pairs NTCIP with IEEE 1588 Precision Time Protocol (PTP) to encompass the fault tolerance and synchronization needed for a distributed Ethernet network. We anticipate that the improved quality of information there are will benefit pedestrians by providing advanced indication of transfer to walk interval, acknowledgement that a request was received from the traffic controller and persistent updates to the countdown timing through the duration of the walk and pedestrian clear interval.

BACKGROUND

The concept behind Smart Signals uses distributed processing architecture for controlling signals and acquiring service requests from pedestrian buttons, vehicle loop detectors, and special service sensors. Internet technology is used for communication between smart devices and the traffic controller. This is in contrast to conventional methods used for traffic control, where signals and sensors used dedicated wires routed from the controller cabinet. One goal of the Smart Signal approach was to lower construction cost by reducing the number of wires that are required for signalized intersections and permit development of new devices by allowing increased complexity of information exchanged between the controller and the signals or detectors.
National Transportation Communications for ITS Protocol (NTCIP)

In the past, each manufacturer of microprocessor based traffic control devices and software either developed or adopted a different, proprietary protocol for data communications. This required extensive integration projects to incorporate different systems from different manufacturers as well as to communicate between systems operated by adjacent agencies. NTCIP provides common standards for protocols that can be used by all manufacturers and system developers to help ease control network assimilation.

A communications protocol defines a set of rules for messaging and how to encode the data contained in those messages. The NTCIP establishes the rules that allow bytes, characters, and strings to be organized into messages that can be decoded by other NTCIP compliant devices. NTCIP is a family of communications standards for transmitting data and messages between microcomputer controlled devices used in ITS. An example of such a system is a computer at traffic control center monitoring and controlling the operation of microprocessor-based roadside controllers at signalized intersections. The computer may send instructions to the traffic signal controllers to change signal timings as traffic conditions change and in return the intersection controllers send status and traffic flow information back to the traffic control center [7].

Simple Network Management Protocol (SNMP) for NTCIP

Since the initial development in 1988, SNMP has developed into the de facto standard for internetwork management. NTCIP recognized the wide use of SNMP and adopted this protocol as a communication standard for use in the ITS industry. Due to its flexibility, it provides management stations the ability to define their message content through the simplicity and robustness of the protocol. Even though there were concerns about the large amount of encoding overhead, it was decided that the protocol provided a core set of functionality and that companion protocols could be developed to reduce large overhead issues.

SNMP is typically applied to managing network devices. Contained within the traffic controller software are managed objects, or variables, that contain parameters that directly relate to the current operation of the intersection. These objects are arranged in a virtual information database, called a management information base (MIB). SNMP allows traffic controllers to communicate portions of their MIB to management centers for the purpose of accessing these objects.

In the Manager-Agent paradigm for SNMP, managed network objects must be logically accessible. Logical accessibility means that management information must be stored somewhere and therefore that information must be retrievable and modifiable. SNMP actually provides the means for retrieval and modification by using a get-set paradigm to exchange individual pieces of data. It is also possible that the traffic controller provide an unsolicited SNMP message that is similar to a get message but driven by an internal event; it is known as a trap.

Each piece of data stored within a device that is accessible via the SNMP protocol is called an object. Objects are organized hierarchically within the MIB as per the rules set in the Structure of Management Information (SMI) protocol [8]. The SMI organizes, names, and describes information so that logical access can occur. The SMI states that each managed object must have a name, syntax, and an encoding. The name, an object identifier (OID), uniquely identifies the object. The syntax defines the data type, such as
an integer or a string of octets. The encoding describes how the information associated with the managed objects is serialized for transmission between machines. SNMP uses a subset of Abstract Syntax Notation One, (ASN.1) [9]. The SNMP compilation rules for encoding data-types into bits and bytes are defined by Basic Encoding Rules (BER). Definition of the MIB conforms to the SMI specified in RFC 1155. The latest Internet MIB is established in RFC 1213 and is called MIB-II [10].

Traffic controller manufacturers compile their MIB using standardized tools. NTCIP standards 1201, Global Object (GO) Definitions, and 1202, Object Definitions for Actuated Traffic Signal Controller Units, contain definitions of standardized objects using ASN.1 notation [10, 11]. Proprietary objects must be defined by the manufacturer but are still defined and compiled in the same standardized manner. The successful completion of a MIB compilation results in generating a text file which provides links to the OIDs of all addressable objects contained within the traffic controller. The text file will be referred to as “OidNamesOut.txt” in this paper. Each OID is written as a sequence of decimal digits separated by periods. This sequence is generally around 17 bytes long when encoded. An object instance is identified by appending the instance number to this base object identifier. Thus, each instance of data within the device has a unique number associated with it.

**Dynamic Objects (STMP)**

The NTCIP technical working group developed the Simple Transportation Management Protocol (STMP) for application layer bandwidth reduction. STMP uses a similar GET/SET paradigm to that of SNMP without the Protocol Data Unit (PDU) overhead of object identifiers and error codes. The content of every data packet requires each protocol entity to have prior knowledge of the configuration of that message. Every message is built from a user defined structured collection of variables known as a dynamic object. The process of building a dynamic object is a runtime operation that requires communication using SNMP and a list of object identifiers included in the MIB. NTCIP dictates that up to 13 dynamic objects can be defined within the traffic controlling device. The size of a dynamic object is limited by the maximum packet size of the communications network[12].

**System Architecture**

In response toward comments pertaining to initial research on Smart Signals concepts from practitioners and reviewers, we have moved our focus to communicating NTCIP for communications between the smart signal devices and the TS2 cabinet, MMU, and traffic controller. Because NTCIP is not currently designed for time critical applications, we applied IEEE 1588 PTP time synchronization for added system reliability and supervision [13].
IEEE 1588 Time Sync Protocol is applied to low level device management to reduce non-determinism, packet collisions, and coordinate transitions by using a common synchronized clock. TDMA, or time division multiple access, is employed to provide each node with a specified time slot to communicate with the master coordinating device. Because the coordinating device can determine which devices are synchronized; it then acts as a distributed control malfunction management unit. Figure 1 demonstrates all common internetwork messages inside the PTP and NTCIP communication stack.

**Smart Signal Network Devices**

The architecture of the Smart Signals system is based on an Ethernet backbone. It is comprised of four critical devices; the network management unit (NMU), pedestrian signals, networked accessible pedestrian stations and a pedestrian smart signal controller.
SSN is an abbreviation for Smart Signal Network which represents all communications to allocate, manage, and control the distributed pedestrian architecture.

The design shown in Figure 2 includes a NMU that implements the MMU type functions for the networked devices. The NMU maintains the master clock and time slot management. It represents an interpretation of how the TS2 standard could evolve into a network based model; it does not represent full TS2 compatibility. The intent is that in the event a node behaves unexpectedly, the intersection would be put into a safe-fail state.

The network accessible pedestrian stations (APS) place pedestrian call and receive the pedestrian signal display status by network communications. This eliminates the need to control wiring between the pedestrian signals and APS that is common practice today. Besides to conventional control and status functionality, the smart devices can communicate the results of self diagnostic tests back to the controller or even the traffic control center.

![Smart Signals Network Architecture Diagram](image)

Figure 2: Smart Signals Integration to TS2 Traffic Controller Cabinet

The smart pedestrian signals display status information received indirectly from the traffic controller. Due to traffic controller device limitations, it is not currently possible to request a dynamic object broadcast from the traffic controller. Therefore, the
pedestrian Smart Signal controller device must request and rebroadcast the NTCIP message.

**Methods and Materials**

**Hardware**

To the extent possible, the hardware used for this investigation is based upon standard industry products to meet Manual on Uniform Transportation Control Devices (MUTCD) requirements and is organized as shown in Figure 2 [14]. The added equipment inside the traffic controller cabinet includes the NTCIP Manager or Ped Smart Signal Controller, PTP malfunction management unit or NMU, an Ethernet switch, and a Ethernet over Power line (EoPL) modem.

We used an Econolite model ASC3-2100 that is a NEMA TS2 Type 2 traffic controller [15]. The Smart Signal Controller and Smart PTP NMU are microcontroller units that use a Rabbit Semiconductor RCM 3000 series microprocessor with a 10 Mbps Ethernet controller [16]. A network switch is needed to communicate Ethernet messages with the ASC3 traffic controller for data retrieval and management. The NTCIP messages must be rebroadcast by the Smart Signal Controller to all smart signals devices connected to the Ethernet over Power-Line (EoPL) network. EoPL modems function as an Ethernet hub distributed over existing power line infrastructure. A single EoPL modem was connected to the switch which basically extends the capacity and ease of connectivity on our local network. Future designs will operate at low AC voltages. The tests reported on for this paper used commercial models that operate at 120VAC.

![Figure 3: Pedestrian signal and button modified for smart signals operation.](image)

Equipment outside the cabinet consists of four Econolite 12 inch polycarbonate
countdown pedestrian signals that were customized as shown in Figure 3 to interface with the smart signal network using Netgear HDX101 200 Mbps EoPL adaptors [17][18]. The proprietary controller is based upon a Rabbit Semiconductor RCM 3000 series microprocessor with 10 Mbps Ethernet controllers. The microprocessor manages signal LED illumination, networking, signal status, and push button activity.

Network Software
The experiment requires a TS2 type traffic controller with a NTCIP compliant MIB. Wireshark [19], a network protocol analyzer program available at no cost, was used to make our observations. Each smart signal node operates on a minimized TCP/IP Ethernet stack.

Communications

User Datagram Protocol (UDP) Addressing, Ports and Size
For this project we choose the UDP/IP Internet Transport Profile for system communications, as defined in NTCIP 2202. This incorporates placing the data stream into an UDP datagram and then placing the UDP datagram into an IP packet. For STMP communications, the NTCIP standard specifies that all communications be directed on port 501, whereas SNMP typically uses port 161.

There is a significant savings in message size of STMP over SNMP. For example, the size of single OID SNMP get-request for one object is 42 bytes. In comparison the STMP request message contains just one byte and can request as many objects as desired. Continuing this example, an SNMP get-response for a single OID of integer type is 43 bytes and the STMP response of that same variable would be three bytes.

The sequence graph shown in Figure 4 demonstrates the communications for the dynamic object data within the smart pedestrian signal network. The device initiating the dynamic object request is IP address 192.168.1.101 and the traffic controller is 192.168.1.5. Once the response is receive by the device initiating the request, it is rebroadcast to all nodes on that network using IP address 255.255.255.255 on a new unique port where all of the Smart Signals pedestrian devices are listening. The broadcast port assignment for the data is arbitrary; however it is critical that the pedestrian devices are all listening to the predetermined port.
Employing Dynamic Objects

The sequence of SNMP commands shown in Table I must be sent to the controller by the client to create a dynamic object. The diagram uses the term “SET” as an abbreviation for an SNMP set operation followed by the variable type in parenthesis. The object name has a corresponding OID which can be found in the OidNamesOut.txt. The sequence begins by setting the configuration OIDs to “Invalid” and then “underCreation” status, so that the new dynamic object can be created. Next the new dynamic object is specified in steps (3) and (4) with the particular OID and the appended dynamic object number desired. Step (4) should be repeated in the order of your desired list of objects using the OID and the incremented value ‘n’ after each set. The sequence is concluded by setting the configuration status to “Valid”.

Table I. Sequence of Steps to Configure a New Dynamic Object

<table>
<thead>
<tr>
<th>SNMP (type)</th>
<th>Object Name</th>
<th>Status (Value)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. SET (int)</td>
<td>dynObjectConfigStatus</td>
<td>Invalid (3)</td>
</tr>
<tr>
<td>2. SET(int)</td>
<td>dynObjectConfigStatus</td>
<td>underCreation(2)</td>
</tr>
<tr>
<td>3. SET(string)</td>
<td>dynObjectOwner.#</td>
<td>“SmartPed”</td>
</tr>
<tr>
<td>4. SET(OID)</td>
<td>dynObjectVariable.#.(1...n)</td>
<td>“1.3.6.1.2.1...”</td>
</tr>
<tr>
<td>5. SET(int)</td>
<td>dynObjectConfigStatus</td>
<td>Valid (1)</td>
</tr>
</tbody>
</table>

Retrieving a Dynamic Object

A device can retrieve a dynamic object by sending one byte of data within a UDP message. Transportation Management Protocol (TMP) defines all dynamic object get
requests to be comprised of the hexadecimal value 0x80 masked with the number of the
dynamic object. A response message contains the header 0xC0 followed by the raw
serialized data. An example query with five OIDs programmed into dynamic object
number one is detailed in Table II.

Table II. Definitions of packet fields for example dynamic object

<table>
<thead>
<tr>
<th>Payload:</th>
<th>Details:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>The request packet of one byte</td>
</tr>
<tr>
<td>81</td>
<td>stmp-get-response for dynamic object #1</td>
</tr>
<tr>
<td></td>
<td>The seven byte response header</td>
</tr>
<tr>
<td>C1</td>
<td>stmp-get-response for dynamic object #1</td>
</tr>
<tr>
<td></td>
<td>The included information field</td>
</tr>
<tr>
<td>14</td>
<td>variable 1 = phasePedestrianClear.2</td>
</tr>
<tr>
<td>02</td>
<td>variable 2 = phaseStatusGroupPedCalls.1</td>
</tr>
<tr>
<td>00 12</td>
<td>variable 3 = asc3PedTimer.1</td>
</tr>
<tr>
<td>02</td>
<td>variable 4 = phaseStatusGroupPedClears.1</td>
</tr>
<tr>
<td>09</td>
<td>variable 4 = phaseStatusGroupPedClears.1</td>
</tr>
</tbody>
</table>

In our observation of the experiment using the Wireshark protocol analyzer,
extaneous data was included in the transmitted response packet from the controller. The
origin of this anomaly comes from how the UDP header information is defined, which in
this case was how the UDP minimum packet size was defined; the extraneous data should
be excluded when decoded.

PTP Communication Scheduling
A global time base generated using PTP synchronization is divided into mutually
exclusive time slots dedicated for data transmission. In our application, we use fifteen
time slots allocated to ten spatially distributed devices. Figure 5 shows the
communication scheduling scheme [20] used for distributed traffic control system
applications. In Figure 5, the non-idle slots are 15.625 ms long and idle slots are 7.813 ms
long. Idle slots are used to allow Smart Signal network devices to process sync messages
and NTCIP update control messages. The summation of all of the slot times results in
each scheduling cycle interval requiring 203.125 ms. Using this scheme, the NMU can
detect critical and non-critical faults within a span of 203 ms.

Slot number zero is dedicated to the NMU which is essentially the PTP master
device. This slot is used to broadcast PTP sync messages to the network devices and
contains necessary information for synchronizing PTP slave clocks. Slots numbered
three through ten are allocated to spatially distributed Smart Signals. During these time
slots, the Smart Signals broadcast their status messages. If a request for service is
initiated from by a pedestrian, the pedestrian set detector NTCIP message is also sent to
the traffic controller within the same slot interval. The pedestrian request is
acknowledged in the NTCIP slot interval when the intersection status broadcast verifies
the pedestrian detector set. Slot number 11 is allocated to the NMU to transmit status messages, such as timing updates. Slot number 12 is allocated to the NTCIP traffic controller timer and status update. During this slot, the Smart Signal Controller sends intersection status messages which could include the flashing condition or information sharing between signals and buttons. After sending the status message the Smart Signal Controller sends an STMP control request message to the TS2 traffic controller and rebroadcasts the received control message to the remaining Smart Signal devices.

Figure 5: Smart Signal PTP Communication Scheduling Scheme

Fault Detection and Handling
Failures occur when a component fails to meet system data integrity or time response requirements. Hardware or software failures within a distributed Smart Signal device that inhibit reliable operation of the overall system cause critical faults. Detection of status message omission helps the system to identify faulty nodes. A device in the active state is responsible of consistently transmitting status messages during its allocated time slot. Failure to receive status messages within the scheduled sender node’s time slot is interpreted as a device fault. Faults within the Smart Signal Controller or the NMU are considered to be critical faults. Occurrence of critical fault triggers the system to enter safe-fail mode. In this mode, all pedestrian signals would turn off as per MUTCD specifications [14]. A failed pedestrian device is treated as a non-critical fault as long as it can be determined that the pedestrian signal in not displaying the walk light. On occurrence of non-critical faults the system maintains normal operation; however there is some loses some of functionality.

Results and Analysis
NTCIP 1103 defines an SNMP or STMP request as timed-out if from the time a request is received and responded to exceeds 100 (ms) plus 1 (ms) for each byte contained in the variable-binding field (OID and data), unless otherwise specified by a communication standard. We will use this reference as we look at SNMP and STMP responses from the ASC3 traffic controller. The experiment compared two SNMP and STMP response packets that are similar in total UDP data size.
Message Timing: SNMP

The timing test results shown in Table III were generated by polling the traffic controller with unique SNMP requests every 200 (ms). The average response time was around 12 (ms) with a maximum of 170 (ms). The experiment duration was 215 minutes. The same test was conducted with a 225 ms polling interval and in general the response times were between 20-56% more consistent and responsive. Unfortunately response times were still erratic as signified by the relative high standard deviation seen for both intervals. However there was an observable trend around four milliseconds and fifty milliseconds response times.

<table>
<thead>
<tr>
<th>Message Timing: STMP</th>
</tr>
</thead>
</table>
| Two request intervals were tested for STMP. The timing test results shown in Table III, were generated by polling the traffic controller with a dynamic object every 225 (ms) and 200 (ms). The dynamic object was constructed to respond with 47 objects which is equivalent in data packet size to the two objects queried with the SNMP tests. With the initial test of 200 ms polling interval, the response delay actually increased at a linear rate as time progressed. The NTCIP defined time-out condition, was reached approximately 200 minutes into the test. At this interval, STMP has a high standard deviation that exceeds that of SNMP. Overall performance was worse because the average and maximum were much higher than desired. In the 225 ms polling interval, the response times were significantly improved above that of SNMP. The average response time was around two milliseconds. The most significant improvement was the standard deviation which was down to $\sigma = 0.00158$ from $\sigma = 0.0625$ and much better than the SNMP at $\sigma = 0.0121$. We found standard deviation at the 225 ms polling interval to fit the behaviour necessary for PTP time slicing.

Advantages and Disadvantages of STMP

When comparing the timing and packet size per object payload, using dynamic objects has significant advantages. The reduced processing for protocol overhead was evident when comparing the SNMP to the STMP response standard deviation along the same polling interval. The SNMP protocol overhead seemed to cause a more erratic response delay within the tested traffic controller. In a time critical environment, such as our PTP constrained smart pedestrian signals, SNMP would need to provide a more consistent reply. STMP proved to be a much more efficient means for extracting our specified pedestrian information from the traffic controller.

The disadvantages of dynamic objects are few, but message integrity and abnormal timing shall be considered. Part of the initial testing of dynamic objects yielded
inconsistent data because of a lack of knowledge for the objects contained in the STMP response. The receiving agent must have prior knowledge of the dynamic object structure because it is not encoded within the message. Response delay was also an issue. The abnormal behavior of the traffic controller at a 200 ms polling period was reason for concern. It is concluded, that in these test conditions instances where a dynamic object is polled at period less than 225 ms for an extended period of time will result in an increasingly undesirable response delay.

**Synchronization Error**

Factors such as clock drift, clock resolution, clock variance, communication latency, and communication delay fluctuations must be taken into account in order to synchronize clocks across an Ethernet network. Each of these factors contributes to synchronization error. Figure 6 shows frequency distribution plot for the percent synchronization error of our system. This plot was obtained by running a synchronization test for 12 hours. During this test, samples were taken from the PTP master and the slave clocks every 500 ms. The clock samples were then compared to generate the data for the plot in Figure 6. One observes that during majority of the test period interval the synchronization error was limited to ±100 µs. The worst case synchronization error for this 12 hour test was 808 µs. It is important to minimize this error in order to develop a communication scheduling scheme with mutually exclusive time slots.

![Synchronization Error Frequency Distribution](image)

**Figure 6: Synchronization Error Distribution Using PTP IEEE 1588**

**Omission Fault Detection**

Figure 7 illustrates the detection of a critical fault that is generated by unplugging the Ethernet cable connected to the Smart Signal Controller. The slot interval signal represents the period in which that device can communicate. The brief pulses on the middle three signals, indicate that the device communicated something for the pulse duration.

The Smart Signal Controller loses its time slot after its Ethernet cable is removed. The rising edge on NMU failure detect signal represents the omission failure detection.
The NMU detects the omission fault within 10 ms because the communication slot in which dynamic object is broadcast was expected shortly after. On detecting this omission failure, the NMU puts the Smart Signal Network into a safe-fail mode of operation. This does not occur until the status update slot which occurs shortly before the Smart Signal detects the fault condition.

![Analysis of Omission Fault on Smart Signal Network](image)

**Figure 7: Analysis of Omission Fault on Smart Signal Network**

**Future Work**

Throughout our development of smart signals concepts, we seriously consider feedback from a diverse group of traffic industry practitioners. In an effort to resolve their concerns we have identified the following topic for further research.

*Extended NTCIP managed communications:* The Smart Signals currently rely on a predefined dynamic object. In the future iterations it would be useful to have an interface that allows for additional configuration of dynamic objects and the managed nodes in the field.

*NTCIP compliant Accessible Pedestrian Buttons:* Current operation has pedestrian button management inside the Smart Signal software. For field testing and future municipal
acceptance of Ethernet networked traffic devices, we believe development of an advanced pedestrian button designed for special needs users to be the most prudent.

Intersection fault modes: For the PTP managed MMU it is necessary to now research more advanced fault conditions beyond the omission fault documented in this paper.

Alternate Traffic Controllers: Although this experiment works well with the Econolite ASC3, in order to achieve industry adaptation, the system should be tested on other (Ethernet enabled) traffic controllers that conform to NTCIP standards.

NTCIP Traps: The implementation of NTCIP defined traps would significantly reduce network traffic and possibly increase reliability. Unsolicited messages from the traffic controller preemting signal and timing transitions would be a next step to integrating Smart Signals into the traffic controller.

Traffic controller operations efficiency supervisor: Development of an independent application that analyzes the efficiency of the traffic timing plans which could be implemented using dynamic objects to extract the information.

CONCLUSION

This is an efficient method for distributing signal control information to pedestrian traffic signals. The overall pedestrian system effectiveness was improved by implementing an Ethernet infrastructure. The system described has the capability for improved communications that can result in more reliable operations and a higher degree of functionality. The experiments demonstrated that low cost microprocessor based devices are readily able to communicate using NTCIP and PTP protocols. Both SNMP and STMP objects were investigated and shown that there was performance threshold with a 200 ms polling interval. The measured delays were unique to the software of TS2 controller selected for the experiment. Given that traffic controllers are designed for real-time operation, we believe that software modifications to any NTCIP compliant controller would lessen the observed delays. STMP has distinct performance advantages over SNMP for managing real-time control because of its inherent protocol efficiency and simplicity. The PTP communication scheduling scheme was integrated with eight Smart Signals which were synchronized and managed as a MMU would within the TS2 standard. The devices were synchronized to ±100 µs. However, future distributed traffic control applications that require more devices may use multiple Smart MMUs for handling multiple clusters without increasing fault detection time. Omission failure was demonstrated and detected within 10 ms and the system could be placed into flash within 203 ms.
ACKNOWLEDGMENT

Funding provided to the National Institute of Advanced Transportation Technology (NIATT) by the U. S. Department of Transportation, Research and Innovative Technology Administration, Grant No. DTRS98-G-0027. Equipment and engineering support was provided by Econolite Control Products, Inc. 3360 East La Palma Ave. Anaheim, CA, 92806 and Campbell Company, 221 W 37th St., Ste. C, Boise, ID, 83704.

References


