
KUONG-HO CHEN, CHYI-REN DOW, CHEN-WEI YANG AND WEI-CHUN CHIANG
Department of Information Engineering and Computer Science
Feng Chia University
Taichung, 407 Taiwan

To achieve automation in Intelligent Transportation System (ITS) environments, it is preferred that integrations for existing traffic management systems as well as various vehicle-centered systems can be done without major modification. This is in particular essential for an automated emergency vehicle preemption system, where traffic signal preemption should be more accurate, smooth, and safe. Current researches for ITS middleware do not give much thought on interoperability and performance. Thus they cannot support such automation framework. This paper proposes Bevor, an ITS middleware for emergency vehicle preemption based on National Transportation Communications for ITS Protocol (NTCIP). By utilizing NTCIP protocols in Communication Layer, Bevor achieves communication and data interoperability across most devices and equipments. The deployment of NTCIP STMF also significantly improved the effectiveness of transmission. The Service Layer of Bevor is designed with the Web 2.0 framework and XML exchange, enabling easy access to data as well as operations such as event detection, policy matching, etc. A prototype system was implemented, and the experimental results show that Bevor offers a great interoperability, more flexible automation control, reuse, and improved performance.

Keywords: intelligent transportation system, emergency vehicle preemption system, NTCIP, web service, STMF

1. INTRODUCTION

Increasing development in vehicle, communication, as well as traffic management technology has led to numerous novel applications in the field of ITS [11, 14]. Among them, Emergency Vehicle Preemption (EVP) systems in particular are beneficial to human lives [8-10]. In many countries, emergency vehicles such as ambulances are given dispensation from traffic regulations such as traffic signal or speed limit. Although these privileges are granted for saving of human life, very often extra accidents were caused during the Emergency Vehicle (EV) transportation, especially in road junctions [1]. In 1997, over 15,000 accidents with emergency vehicles racing to emergency sites occurred in USA. EVP systems were therefore designed to improve EV safety/efficiency by providing prioritized road accesses [5]. Traditional non-ITS technologies include strobe emitters, light emitters, or siren detectors which have inherited problems such as blocked line-of-sight, noise [9], and can be easily duplicated and illegally used by non-EV vehicles [17].

Received March 31, 2009; accepted September 30, 2009.
Communicated by Chih-Yung Chang, Chien-Chung Shen, Xuemin (Sherman) Shen, and Yu-Chee Tseng.
* This paper was supported by the MOE Human Resource Program for ICT, Taiwan, 2008.
Modern ITS proprietary transit preemption solutions such as [2-4, 18] are primarily used for prioritized bus road right. They use either roadside sensors with on-vehicle tags to achieve Selective Vehicle Detection (SVD), or GPS reported through GSM-USSD or 3G network to acquire Automatic Vehicle Location (AVL). The traffic management center gets the location of the buses requesting for preemption, predicts its estimated time of arrival (ETA), and grant as the system sees fit due to traffic conditions or mission type, adapting various strategy for preemption, and configures the traffic signs accordingly by dispatching commands from the center to the field devices [7]. Mostly designed for public transportation, they provide fleet managing functions such as bus scheduling, dispatching, monitoring, as well as traffic controlling functions such as signal monitoring, actuating and the preemption, which generally fixed and closed. These systems, however, either do not explicitly support additional EV preemption, or would require modification to do so. Furthermore, the interoperation between them and other systems of different brands such as Advanced Traffic Management System (ATMS), Advanced Traveler Information System (ATIS), Commercial Vehicle Operation System (CVOS), Fleet Management System (FMS), or Emergency Management Services (EMS) are either non-existing, or require major modification and maintenance, which again cost more efforts and resources. In addition, they may also require additional equipments installed on the EVs, while current EVs usually have tracing equipment such as GPS installed for their own centers to monitor them, causing redundancy and extra cost. Thus, they are not suitable for adapting as base for an ideal EVP system.

ITS-based open middleware architectures such as [6, 11, 12, 14] have been proposed to remedy the problems of interoperability, providing modeling of entities and messages to foster quick development of ITS applications such as accident-detection, in-car advertisement, personalized driver assistance, information service, etc. However, these systems did not give much thought on the compatibility issue, using in-house designed protocols, data models, which might themselves be incompatible with existing traffic infrastructures. If they are not able to get data from sensors as well as actuating devices, on the side of practicality, they cannot quickly support the construction of an EVP system.

This work proposes Bevor, an NTCIP-based interoperable framework for emergency vehicle preemption system using Web service [16] and Simple Transportation Management Protocol (STMP). In the Bevor framework, Web services are used in a traffic management center to create a “window” to accept preemption requests sent from various fleet management centers in Simple Object Access Protocol (SOAP) messages. The traffic management center would then dispatch the proper controlling commands to traffic signs using STMP for the preemption to take effect. Bevor has the following merits:

1. Interoperability: the traffic signal priority system that can now handle the preemption for various vehicles, including buses, EVs, or even railroads as long as the monitoring sides have the ability to call the Web service.
2. Modularity: the traffic management center does not need to keep the jobs of vehicle monitoring anymore, as it is now handled by the fleet management center of the owners of the EVs (who were already doing so today). Software modifications on both sides would not affect the preemption function as long as the service protocol is fol-
The EVs also would not need additional equipments/resources for transmitting data to both sides.

(3) Efficiency: the controlling of the traffic signs are done through the use of Simple Transportation Management Framework (STMF) [15] instead of traditional Simple Network Management Protocol (SNMP), which cuts many redundant transmissions, thus can significantly improve the effectiveness of center-to-field transmission. The establishing of center-to-center communication also contributes to more effective transmission since they are usually in wired environments. These make Bevor architecture able to react more dexterously to preemption requests, and thus require less ahead distance for vehicle detection.

The rest of this work is organized as follows. Related work is described in section 2. The system architecture of Bevor is presented in section 3. System implementation and prototype are described in section 4. Experimental results are shown in section 5. Finally conclusions are made in section 6.

2. RELATED WORK

NTCIP is a series of standards designed to achieve interoperability and interchangeability among electronic traffic control equipments from different manufacturers. NTCIP offers DATEX-ASN (Data Exchange in ASN) and CORBA (Common Object Request Broker Architecture) as application level protocol choices. Primary purpose of NTCIP is to achieve interoperability and interchangeability among units in ITS. NTCIP focuses on two types of communications: C2C (center-to-center) and C2F (center-to-field). In this paper, some message sets and objects were used from NTCIP 1202 which detailed the data objects of actuated traffic signal controller, and NTCIP 1211 which detailed the standard data objects of signal control and prioritization.

In an EVP system there would be much overhead when controlling signs using traditional SNMP since it has large packet header size and can only transfer one object at a time. Many MIB objects would be repetitively sent when using SNMP, thus increases the overhead of the system. STMP is the management protocol defined in [15], and it is developed to decrease redundant data packets of normal SNMP. STMP has been designed to work with dynamic objects defined at run time.

In STMP, as the system boots up, an initial setup of dynamic object definitions would be executed. In subsequent transmission, the definitions of objects would not be needed again. Only the number of desired objects and the new values to set are required, which is very often only several bytes, and up to 255 objects could be transferred at once. While greatly cutting the packet size, STMP also has the benefit of providing the management station with the flexibility required to define its own messages, providing a potentially significant advantage in applications involving frequent polling on limited bandwidth links.

Meanwhile, a Web service is a software system designed to support interoperable machine-to-machine interaction over a network [16], which is adapted for C2C method for Bevor. Web services are essentially application programming interfaces (API) executed on a remote system hosting the requested services, and accessible through the Internet. Such services use Extensible Markup Language (XML) messages that follow
the SOAP standard and have been popular with traditional enterprise. In such systems, there is often a machine-readable description of the operations offered by the service written in the Web Services Description Language (WSDL), which is required for automated client-side code generation in many Java and .NET SOAP frameworks.

3. SYSTEM ARCHITECTURE

This section describes the system architecture of Bevor, its components, and its communication specifications.

![Bevor System Architecture](image)

**3.1 Bevor Framework**

The architecture of Bevor is shown in Fig. 1. The context of Bevor $B_0$ is defined as \( \{T_M, F_M, V, D\} \), where:

- \( T_M \) denotes the traffic management system, usually installed in terminal rooms in local traffic administration departments. In Bevor, \( T_M \) has minimally two components: an EVP system and a sign actuating system. The EVP system has permanent Internet connection, and hosts one or more Web services to accept requests for traffic road right preemption. \( D \) denotes the set of field devices in the context. These can be traffic light controllers, dynamic message signs, ramp meter controllers, or other devices with ability to process STMP messages, and the ability to operate according to its respective standards of the NTCIP family. The sign actuating system in \( T_M \) has permanent connections with all \( dv_e \in D \), and constantly polls their status.

- \( F_M \) denotes a fleet management system, installed in emergency management centers, police departments, or bus dispatching centers. It has knowledge of the map complex in which its fleet operates, and should be able to identify the locations of road junctions,
railroad crossways, highway entrances, or other junctions that may require prioritized traffic. It also maintains a database recording position, speed, messages, and other information of each vehicle. Being a real-time fleet management system, $F_M$ has either constant or periodical connections with its managed vehicles $V$. Each $v_i \in V$ is equipped with mobile data terminal equipments that enable it to maintain such connection with the center, as well as providing GPS capabilities. Currently the trend of such mobile network is handled by an ISP through the RDS, GSM, SMS, or GPRS systems. The choice of which depends on the balancing between affordability and data packet frequency. Furthermore, $F_M$ has the ability to form SOAP message, and has constant access to the Internet so that it could send the SOAP message to $T_M$ in order to invoke the EVP service.

3.2 Preemption Procedure

The preemption procedure is divided into four phases, described as follows:

1. Initialization phase: Once $F_M$ gives a mission to $v_i \in V$, $F_M$ would request the mission to be authenticated by $T_M$. $T_M$ would then authorize the right to preemption using a cryptographically secure pseudo-random number generator (CSPRNG) to issue a unique key $k_j$ to $F_M$.

2. Monitor phase: Once $v_i$ moves onto the road in the emergency state, it would constantly report its GPS location to $F_M$. In this phase, $F_M$, installed with fleet management system, would be able to ascertain if $v_i$ approaches a road junction. Once inside a distance threshold ($T_{SVD}$) from the junction, $F_M$ would send $k_j$, the location of $v_i$, and the ETA to $T_M$ thus moving into the next phase.

3. Preemption phase: Upon receiving preemption request from $F_M$, $T_M$ locates the involving traffic light devices $d_{v_i} \in D$, and then sends $d_{v_i}$ a control code for the preemption to take effect. Then, $d_{v_i}$ changes the states of traffic signals accordingly, and notifies $T_M$, which in turn would notify $F_M$. The driver of $v_i$ checks traffic signal state and then passes through the road junction. After the desired time window, $T_M$ would send control command to restore the status of $d_{v_i}$. At the end of the preemption phase, the procedure would go again to the monitor phase. In each junction it would enter the preemption phase again; this would repeat until $v_i$ reaches its destination, at which point the procedure enters the arrival phase.

4. Arrival phase: when $v_i$ arrives at destination, the signal priority control process is done. $F_M$ would send a message to $T_M$ to register off $k_j$ so that it would no longer function.

The following sub-functions are predefined for depicting the pseudo code of the procedures of preemption:

- $\text{ReachedDest}(v_i)$ is executed by $F_M$ to check whether $v_i \in V$ has arrived at its destination, and then returns a Boolean value as result.
- $\text{ApproachJunction}(v_i, T_{SVD})$ is executed by $F_M$ to determine if $v_i$ has come near to a junction. The proximity is determined by a SVD threshold $T_{SVD}$, and then returns a Boolean value.
- $\text{GetLocation}(v_i)$ is executed by $F_M$ to poll from $v_i \in V$ its location, and then returns a set of location as result.
Fig. 3. EVP Pseudo-code of a traffic management center.

- **GetDevice(vi)** executed by $T_M$ to determine the next device controller vehicle $v_i$ is about to pass, and then returns the device’s ID number as result.
- **SendMessage(destination, message)** is used by both $T_M$ and $F_M$ for C2C communications. The message formats are defined in the next section.
- **ReceiveMessage(message)** is used to listen for a message, and then returns a Boolean value.
- **EstimateTouch(vi)** is used to estimate the ETA of $v_i$ to the next junction location, and then returns a time value $t_{ij}$.

Figs. 2 and 3 show the execution processes of $F_M$ and $T_M$, respectively in the form of pseudo code, presented using above definitions and phases. The procedure of $F_M$, shown in Fig. 2, starts as it prepares $v_i$ for an emergency mission and concludes when $v_i$ has reached the destination. It is idle in the preemption phase, as $v_i$ waits for the signs to change in order to pass the junction. The procedure of $T_M$, shown in Fig. 3, would constantly check for new preemption requests. It is idle in the monitor phase, since the monitoring is done by $F_M$ and $T_M$ to simply listen for requests.

### Procedure 1: Fleet Management Center

```
Begin
  // Initialization phase
  FM.SendMessage(TM, MSG_SK)
  // Monitor phase
  WHILE(!ReachedDest(vi)) DO
    GetLocation(vi)
    if(ApproachJunction(vi, TSVD)) THEN
      EstimateTouch(vi)
      FM.SendMessage(TM, MSG_PE)
    END IF
  END WHILE
  // Preemption phase
  // waits for vi to pass
  // Arrival Phase
  FM.SendMessage(TM, MSG_DK)
End
```

Fig. 2. EVP Pseudo-code of a fleet management center.

### Procedure 2: Traffic Management Center

```
Begin
  WHILE(1): DO
    // Initialization phase
    if(ReceiveMessage(MSG_SK)) THEN
      IF(CheckAuth(MSG_SK)) THEN
        MSG_IK = CSPRNG(vi, FM)
        SendMessage(FM, MSG_IK)
      ELSE
        Set AUTH_FAILED
      END IF
    END IF
    // Monitor phase
    // waits for vi to arrive
    // Preemption phase
    IF (TM.ReceiveMessage(MSG_PE)) THEN
      dv = TM.GetDevice(vi)
      TM.SendMessage(dv, MSG_PR)
      If(ReceiveMessage(MSG_PRRes)) THEN
        TM.SendMessage(FM, MSG_PERes)
        wait(tij)
        TM.SendMessage(dv, MSG_PC)
      END IF
    END IF
    // Arrival Phase
  END WHILE
End
```

Fig. 3. EVP Pseudo-code of a traffic management center.
3.3 Communication Messages

The messages transmission process is shown in Fig. 4. In NTCIP standards, objects and messages are composed of more elemental objects. Centers then use these messages to control devices on the roadside. Since there are only objects about signal priority control in the standard, to pass the information of vehicles in recognizable form, the objects concerning EVP system must be defined. There is currently only one message set, called Weather Report Message Set for Environment Sensor Station in NTCIP, which is not much help. Thus for EVP, control message sets must be defined by combining other NTCIP objects. There are four pairs of messages in EVP and Signal Priority Control message sets proposed. Table 1 shows each message defined using NTCIP MIB objects, and they are described as follows:

1. **MSG_SK**/ **MSG_IK**/ **MSG_DK**: **MSG_SK**, with **DEAUTHENTICATION** = 0, is used by **FM** as a request to **TM** to get a key. **TM** then replies **FM** with **MSG_IK** which includes the key. To return a key, **FM** would send **MSG_DK** with **DEAUTHENTICATION** = 1 to **TM**.

2. **MSG_PE**/ **MSG_PERes**: **MSG_PE** is a preemption request from **FM**, and **MSG_PERes** is a response of **MSG_PE**.

3. **MSG_PR**/ **MSG_PRRes**: **MSG_PR** is a controlling code to change. **MSG_PRRes** is a response of **MSG_PR** when **dvr** finishes setting the states.

4. **MSG_PC**/ **MSG_PCRes**: After **v_i** passes through intersections, **MSG_PC** is used as a command by which **TM** informs **dvr**, to restore the states of traffic signals before pre-emption. **MSG_PCRes** is a response of **MSG_PC** while **dvr** finishes setting the states.
Table 1. Message set.

<table>
<thead>
<tr>
<th>MSG_SK/MSG_IK/MSG_DK</th>
<th>MSG_PE/MSG_PERes</th>
<th>MSG_PR/MSG_PRRes</th>
<th>MSG_PC/MSG_PCRes</th>
</tr>
</thead>
<tbody>
<tr>
<td>priority request</td>
<td>secret key</td>
<td>priority request</td>
<td>priority request</td>
</tr>
<tr>
<td>request center id</td>
<td>priority request</td>
<td>request center id</td>
<td>center id</td>
</tr>
<tr>
<td>priority request</td>
<td>priority request</td>
<td>priority request</td>
<td>priority request</td>
</tr>
<tr>
<td>vehicle id</td>
<td>center id</td>
<td>vehicle class type</td>
<td>center id</td>
</tr>
<tr>
<td>global time</td>
<td>priority request</td>
<td>priority request</td>
<td>priority request</td>
</tr>
<tr>
<td>secret key</td>
<td>vehicle class type</td>
<td>vehicle class level l</td>
<td>vehicle class type</td>
</tr>
<tr>
<td>deauthentication</td>
<td>priority request</td>
<td>priority request</td>
<td>priority request</td>
</tr>
<tr>
<td>vehicle longitude</td>
<td>time of service</td>
<td>device strategy number</td>
<td>device strategy number</td>
</tr>
<tr>
<td></td>
<td>desired</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>time of estimate departure</td>
<td></td>
</tr>
<tr>
<td>priority request</td>
<td></td>
<td>time of estimate departure</td>
<td></td>
</tr>
<tr>
<td>vehicle latitude</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>priority request</td>
<td></td>
<td>time of estimate departure</td>
<td></td>
</tr>
<tr>
<td>time of estimate</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>departure</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

4. SYSTEM IMPLEMENTATION

This section describes our system implementation and prototype, the goals of this system implementation are interoperability and interchangeability to reach emergency vehicle through intersection quickly and safely. The field side represents traffic lights, and service centers use cable to connect. The service centers and emergency vehicles can be connected by various wireless systems, such as GPRS, 3G, WiMAX, 802.11p, etc. The Bevor middleware protocol stacks are shown in Fig. 5. In the transport and subnet-work levels, the protocol stacks are constructed by the Windows socket library and related plant LAN access profiles. The IP (Internet Protocol) is used widely for various applications to connect to the Internet. The UDP can support various data exchange in the Internet and more suitable than TCP in wireless connection. The system in the transport level uses UDP and acknowledgement mechanism that can reach data exchange more reliable.

In the application level, for reducing communication session times, STMP is used in transportation management protocol to replace SNMP. STMP supports predefined and runtime defined objects, called dynamic objects. The emergency vehicle uses Emergency Data eXchange Language Approximation EDXLA (EDXLA) to send emergency data.

The traffic management center is a Microsoft Windows 2003 server, and the Web service is implemented using IIS 6.0 ASMX. The fleet management center is a modification from on-shelf logistic fleet management system, changed so that it would invoke the
AN NTCIP-BASED INTEROPERABLE FRAMEWORK FOR EMERGENCY VEHICLE PREEMPTION

Applications
Twisted pair
Ethernet
UDP
MIBs
SNMP
Dynamic
Object
STMP
Wireless
(802.11a,GPRS)

Information Layer
Application Layer
Transport Layer
Subnetwork Layer
Plant Layer
LAN Access Profile
UDP
Ethernet,
EDXLA
MIBs
EDXLA
EDXLA
EDXLA
EDXLA

Fig. 5. Bevor protocol stack.

POST /WebService/PreemptionService.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://mclab/preemptionIn"

<?xml version="1.0" encoding="utf-8"?
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"

<soap:Body>
<MSG_PE xmlns="http://www.dummy-tmdd-address">
<CenterID xmlns="">MCLAB</CenterID>
<VehicleClassType xmlns="">ambulance</VehicleClassType>
<VehicleClassLevel xmlns="">1</VehicleClassLevel>
<VehicleLongitude xmlns="">120.6487</VehicleLongitude>
<VehicleLatitude xmlns="">24.1812</VehicleLatitude>
<TimeOfSD xmlns="">10</TimeOfSD>
<TimeOfED xmlns="">10</TimeOfED>
</MSG_PE>
</soap:Body>
</soap:Envelope>

Fig. 6. SOAP message of a preemption request.

Web service using the IP of the traffic management center. Shown in Fig. 6 is an example of the SOAP message of the preemption request.

For C2F communication, a SNMP/STMP dynamic object transceiver has been designed on the traffic management center as shown in Fig. 7 to configure NTCIP traffic signs.

Bevor Preemption Request Server v.0.09
Initializing devices...done.
Device status update...done.
Checking Web service status...done.
[READY]>

POST /WebService/PreemptionService.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://mclab/preemptionIn"

<?xml version="1.0" encoding="utf-8"?
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"

<soap:Body>
<MSG_PE xmlns="http://www.dummy-tmdd-address">
<CenterID xmlns="">MCLAB</CenterID>
<VehicleClassType xmlns="">ambulance</VehicleClassType>
<VehicleClassLevel xmlns="">1</VehicleClassLevel>
<VehicleLongitude xmlns="">120.6487</VehicleLongitude>
<VehicleLatitude xmlns="">24.1812</VehicleLatitude>
<TimeOfSD xmlns="">10</TimeOfSD>
<TimeOfED xmlns="">10</TimeOfED>
</MSG_PE>
</soap:Body>
</soap:Envelope>

Fig. 6. SOAP message of a preemption request.
5. SIMULATION RESULTS

Experiments were conducted using a testing platform developed using Microsoft Visual C#. The virtual environment is a 5km straight road with 10 road junctions, each installed with preemption-enabled traffic light devices. Normal vehicles were programmed to arrive as Poisson distribution. A single EV is released at the start of the road, and moves with three speed variations: 40kph, 60kph, and 90kph. If the immediate traffic sign ahead of the EV is green, then the EV could bypass other vehicles at though they moved away to allow passage. If red, which means that either the preemption is not triggered in time, or is triggered too early for the EV to catch up, then the EV cannot bypass other vehicles ahead of it, as though they are blocked tightly from the junction and cannot maneuver to allow the EV to pass.

The traffic management center is a preemption server with preemption Web service opened, and is linked to STMP/SNMP capable traffic lights in the virtual environment using real-life networks with two capacities, which are: (1) 56Mbps IEEE 802.11a wireless networks; and (2) 100Mbps wired Fast Ethernet. The fleet management center is a preemption request server, which constantly receives location report from the EV in the virtual environment every three seconds.

The expected result of the experiment is to show improved preemption efficiency as well as response time. To such end, Bevor is compared with the method of iBUS [4] and its own SNMP version. The iBUS scenario was simulated using the following transmission sequence, as was depicted in [4]: \( v_r \rightarrow \text{GPRS} \rightarrow F_{tr} \rightarrow \text{GPRS} \rightarrow v_r \rightarrow \text{RF} \rightarrow tv_r \rightarrow \text{SNMP} \rightarrow T_{tr} \rightarrow \text{SNMP} \rightarrow dv_r \). As shown in Figs. 8 to 10, SVD refers to the triggering distance of the preemption, which is a parameter for selective vehicle detection. For example, if SVD equals 50m, the preemption would then be triggered when the EV approaches into 50m from the next road junction. SVD = N/A refers to the travelling without preemption support. Larger SVD indicates greater chance the EV can successfully arrive within the requested preemption time. However, larger SVD may also bring more damaged road rights to other vehicles. The goal of Bevor is to find a suitable SVD such that it allows fastest passage of the EV while does not bring unnecessary delay time to other vehicles. Each following figure is drawn from the average of 100 runs, with travel time indicating the time needed for the ambulance to complete the total 5km path.

When the increasing of the SVD does not further decrease the travel time, the SVD is then called the Ultimate SVD (USVD) of an EVP system in that particular scenario, and the final travel time is called Ultimate Travel Time (UTT) whose difference with the
original travel time, \( TT\text{_{n/a}} \), is the Ultimate Gained Time (UGT).

The effectiveness of an EVP is defined here as:

\[
pef(SVD) = \left( \frac{\text{Time Gained}}{(TT_{\text{n/a}})_{\text{UGT}}} \right) \%
\]

Thus \( pef(USVD) \) is always 100%. Reaching \( pef(SVD) = 100\% \) with less USVD indicates less delay for other vehicles, thus less damage of overall road rights. In Fig. 8, for an EV with 40kph, it could be seen that when in wired networks, before reaching USVD, the travel time of Bevor with STMP is averagely superior to Bevor with SNMP by 5.2 seconds, and iBUS by 15.3 seconds. Most significant difference occurred in \( SVD = 100\text{m} \), where Bevor with STMP is superior to iBUS by 30 seconds. For USVD, the USVD of Bevor with STMP is 150m, SNMP 150, and iBUS 200m, where both Bevor scenarios are 50m shorter than iBUS’s 200m. In the wireless environment, Bevor with STMP is averagely 10.1 seconds faster than Bevor with SNMP, and 18.8 seconds faster than iBUS. For USVD, Bevor with STMP is not impacted by switching to a less-bandwidthed, while Bevor with SNMP and iBUS both need extra 50m to reach the UTT. In Fig. 9 (a), for an EV with 60kph under wired environment, the USVD of Bevor with STMP is 200m, SNMP 250m, and iBUS 300m. In Fig. 9 (b), only iBUS is impacted by the cut of bandwidth with additional 50m in UVD. Bevor with STMP is superior to both for about 11.4 seconds and 29.6 seconds, respectively.
In Fig. 10 (a) for an EV with 90kph under wired environments, the USVD of Bevor with STMP is 300m, SNMP 350m, and iBUS 500m. In wireless environments, shown in Fig. 10 (b), the UVD of both Bevor scenario is unchanged, while iBUS requires extra 100m. From these results it could be gathered that Bevor architecture with STMP does indeed provide more response time for the EVP to work, which can be seen both from the faster travel time, and from the shorter USVD. This also implicates less sacrificed road rights of common vehicles, thus making Bevor more acceptable by the public.

Note that the number of vehicles in the experiment was generated using Poisson distribution, and more normal vehicles would only serve to negatively impact the travel time when an ambulance failed to pass the intersection in the preemption mode and end up waiting the red light, in which case it would be trapped behind many normal vehicles queued up in front the red light. More vehicles would also bring more damage to the road rights, as more vehicles originally ought to pass the intersections would then have to wait for the preempted sign. It is expected that in cases of road with high density, the SVD required would be lengthened, thus indicating that more normal vehicles would sacrifice their road right.

In Fig. 11, the response time refers to the time from the traffic center sending MSG_PR message to the sign controller, to the traffic center receiving the MSG_PRRes reply from the signal controller. There are the same six MIB objects with same values in both wired and wireless networks.

Fig. 10. Travel time/SVD distance distribution comparison with iBUS (EV Speed = 90 kph).

Fig. 11. STMP/SNMP data object comparison.
MSG_PR and MSG_PRRes messages. In the SNMP protocol, a traffic center or a signal controller can send only one object at a time, both the total SNMP packet size of MSG_PR and MSG_PRRes messages are 554 bytes. However, a traffic center or a signal controller can send at most 255 objects in STMP protocol at a time, both the total STMP packet size of MSG_PR and MSG_PRRes messages are 15 bytes.

In Fig. 11, the comparison of average response time with various amounts of MIB objects between SNMP and STMP can be seen. In the wired environment, the average response time of each object for the SNMP protocol is about 296ms, and adding one MIB object would increase the average response time about 296ms. However, for the STMP protocol, regardless of the amount of MIB objects, the average response time is always roughly about 260ms. In the wireless environment, the average response time of each object for the SNMP protocol is about 610ms, and adding one MIB object would increase the average response time about 600ms. For STMP protocol, regardless of the amount of MIB objects, the average response time is always roughly about 600ms. These showed that STMP, though its response time is doubled by lesser network, still demonstrates significant robustness and scalability in comparison with SNMP.

6. CONCLUSIONS

This work proposes Bevor, an NTCIP-based interoperable framework for emergency vehicle preemption system using Web service and STMP. In the Bevor framework, Web services are used in a traffic management center to accept preemption requests sent from various fleet management centers in SOAP messages. The traffic management center would then dispatch the proper controlling commands to traffic signs using STMP for the preemption to take effect. By using Web service and adapting NTCIP-based MIB objects for STMP message passing, Bevor framework has the merits of modularity and interoperability. Experiment results demonstrate that Bevor is more robust than current methods, being able to function in lesser network communications. Also its reactivity is more agile, requiring less triggering distance for vehicle detection, thus brings less impact to other non-EV vehicles during the preemption.

REFERENCES

5. N. Levick, “Hazard analysis and vehicle safety issues for emergency medical service vehicles: Where is the state of the art?” in Proceedings of the American Society of


Kuong-Ho Chen (陳光和) was born in 1980. He received the B.S. and M.S. degrees in Information Engineering from Feng Chia University, Taiwan, in 2003 and 2005 respectively. He is currently a Ph.D. student in Department of Information Engineering and Computer Science, Feng Chia University, Taiwan. His research interests include mobile computing, intelligent transportation system, intelligent agents, and vehicular ad hoc network.
Chyi-Ren Dow (竇其仁) was born in 1962. He received the B.S. and M.S. degrees in Information Engineering from National Chiao Tung University, Taiwan, in 1984 and 1988, respectively, and the M.S. and Ph.D. degrees in Computer Science from the University of Pittsburgh, U.S.A., in 1992 and 1994, respectively. Currently, he is a Professor in the Department of Information Engineering and Computer Science, Feng Chia University, Taiwan. His research interests include mobile computing, ad-hoc wireless networks, telematics, and learning technology.

Chen-Wei Yang (楊政衛) was born in 1975. He received the M.S. degrees in Information Engineering and Computer Science from Feng Chia University, Taichung, Taiwan, in 2004. Currently, he is a currently Ph.D. student in the Department of Information Engineering and Computer Science, Feng Chia University, Taiwan. His research interests include embedded systems, wireless sensor network and vehicular ad-hoc network.

Wei-Chun Chiang (江惟均) was born in 1984. He received B.S. and M.S. degrees in Information Engineering from Feng Chia University, Taiwan, in 2007 and 2009 respectively. His research interests include mobile computing, intelligent transportation system, and intelligent agents.