Design and Implementation of Safe & Intelligent Bridges System Based on ALE-Compliant RFID Middleware in USN

Jae-Bong Yoo, Byung-Ki Kim, Ho-Min Jung, Taewan Gu, Chan-Young Park, and Young-Woong Ko

***Abstract*—**For accident surveillance and traffic flow conditions, CCTV cameras are installed on bridges. However, their effectiveness is diminished remarkably in a thick fog. Despite heavy fog, our proposed system can detect a car crash on the bridges and propagate accident information to other drivers as quickly as possible to reduce the accident rate and casualties. We designed our system not only to detect car crashes and propagate accident information to reduce accident rate, but also to acquire traffic information and atmospheric information. The Safe & Intelligent Bridges System collects RFID (Radio Frequency Identification) Tag data by using RFID middleware called ALE (Application Level Event). Its interface provides independence between infrastructure components that acquire the raw EPC data, architectural components that filter and count the data, and applications that use the data. This allows changes in one without requiring changes in the other, offering significant benefits to both providers and end-users. It is also able to work together with the USN (Ubiquitous Sensor Network) to implement the ITS (Intelligent Transport System) based on event-driven development framework. We show that our system in the RFID environment provides a novel way to manage traffic systems safely and efficiently by complementing problems in the current traffic system.

***Keywords*—**ALE Middleware, Event Detection, RFID, USN.

I. Introduction

SINCE the twentieth century, transportation is one of the great industries. Based on statistics in Korea, the number of car registrations jumped from around 18,000 in 1955 to 10 million during the late 1990s, and the number was 15.4 million by 2005. The technology of automobiles has been improving so that the number of accidents decreases. Nonetheless, speeding on a highway causes accidents. Consequently, the number of accidents which result in death increases steadily. To reduce the accident rate, ITS (Intelligent Transportation System) could be one of the most suitable solutions [1-3].

ITS technologies are a collection of technologies that increase the efficiency and safety of public transportation systems and offer users greater access to information on system operations [4]. ITS is gradually getting popular nowadays as congestion, traffic accidents, air pollution and so on are becoming critical issues in modern societies. Although it provides information about the state of the road to the appropriate people reliably, accurately, and in a timely manner, ITS in Korea is still the first step [5]. For example, cameras can not detect a car accident when there is a thick fog on a bridge or heavy smoke in a tunnel. Regardless of the surrounding environment such as fog or smoke, RFID (Radio Frequency Identification) is able to show it ability to detect a car crash. RFID is an automatic identification technology that can be used to provide electronic identity to an item or object [6-8].

In this paper, we are focusing on traffic surveillance to detect accidents on a bridge. We propose Safe and Intelligent Bridges based on ALE-compliant RFID middleware without necessarily having to physically alter existing infrastructure to avoid additional collisions.

The organization of this paper is as follows: In Section 2, we first introduce Related Works. Then in Section 3 we present a System Overview, including system components and system architecture. Next, we propose Event Detection Algorithm to detect a car accident in Section 4. Experimental Environment is discussed in Section 5, followed by Application Level Event Simulator and Conclusion.

II. Related Works

Intrusive sensors include inductive loops, magnetometers, and micro loop probes. Generally, these devices are installed directly on the pavement surface by tunneling under the surface. The drawbacks to their use include disruption of traffic for installation and repair, and failures associated with installation in poor road surfaces. Also, sensors are rarely used these days because of their accuracy [9].

Non-intrusive sensors include video image processing, microwave radar, and laser radar. Video-based vehicle detection is a typical solution for traffic surveillance. It has played an important role in real-time traffic management systems over the past decade. Video-based traffic monitoring systems offer a number of advantages over traditional methods, such as loop detectors. In addition to vehicle counting, more traffic information can be obtained by video images, including vehicle classifications, and lane changes. Furthermore, video cameras can be easily installed and used in mobile environments. However, cameras can not detect a car accident

Jae-Bong Yoo, Byung-Ki Kim, Ho-Min Jung, Chan-Young Park and Young-Woong Ko are with Department of Computer Engineering, Hallym University, South Korea (e-mail: {yoojaebong, bkkim, chorogiy, cypark, yuko}@hallym.ac.kr).

Taewan Gu is with Department of Information Technology, Gangwon Human Resource Development Institute, South. Korea (e-mail:gutaewan@korcham.net).

when there is a thick fog on a bridge or heavy smoke in a tunnel [10].

For example, twenty-nine vehicles were involved in a chain-reaction collision on Seohae Grand Bridge on October 4, 2006 leaving 11 people dead and another 54 injured. Authorities blamed excessive speed for the extent of the carnage, but this bridge was enveloped in heavy fog when the accident happened. There was a posted speed limit of 110Kph, but visibility was reduced to an estimated 60-80Kph. The area near the bridge is fog-shrouded in the morning about 30-50 days per year. Comparing the left picture with the right one in Figure 1, CCTV camera on bridges could not detect the car accidents because of a thick fog. When a car is running at the speed of 60Kph and the reaction time is one second, thinking distance is about 16m. In this case, a car accident can not be prevented [11].

Fig. 1 Visibility range of Seohae Grand Bridge (left) and visibility range of regular roads (right)

Authorities should introduce a new system which can detect a car accident despite heavy fog. So, we designed and implemented the Safe & Intelligent Bridges System which can detect a car crash on bridges and propagate accident information to other drivers as quickly as possible to reduce the accident rate and casualties.

III. System Overview

Our system chiefly consists of the seven components as illustrated in Fig. 2: ITS Server, ITS Client, MCU (Micro Control Unit), Zigbee, RFID Reader, RFID ALE Server, and RFID Client. Assuming that RFID readers can read RFID Tags which are attached to cars moving on the road, the RFID Tag on the ticket is issued at toll gates.

First, the ITS Server (①) collects tag information, atmospheric information such as temperature, humidity and illumination etc. The main responsibility of the server is to detect and propagate car accidents. Secondly, the ITS client (②) enables to help administrators manage their system at ease. It processes messages from the ITS server and displays traffic and atmospheric information on the screen. If any message is considered as a car accident, it notifies a police station or a fire station of an accident, providing emergency medical services and technical rescue for drivers. It can send the server requests for controlling lights, sirens, alerts, gates and LCD in a state of emergency as well. The ITS client cooperates with an ALE client (③) which defines ECSpec. Thirdly, the MCU (④) is able to control lights, sirens, alerts, gates and LCD when the ITS server detects accidents and gives commands to MCU. We made the MCU as an embedded board. Fourthly, each Zigbee Sensor node with a RFID reader sends tag information to a Zigbee sink node by using RF (Radio Frequency). Zigbee (⑤) protocols are intended for use embedded application requiring low data rates and low power consumption. A Zigbee node is divided into a sensor node and a sink node as a role: The sink node collects information from a set of sensor nodes and can sense the event such as temperature, humidity and illumination and so. Fifthly, RFID Readers (⑥) read RFID tags and sends them to a Zigbee module by using UART (Universal Asynchronous Receiver and Transmitter). Finally, RFID ALE server (⑦) is communication interface; after the ALE server collects and filters EPCs which attached to cars, it sends them to the ITS server. It can also register logical readers for flexibility and scalability [12, 13].

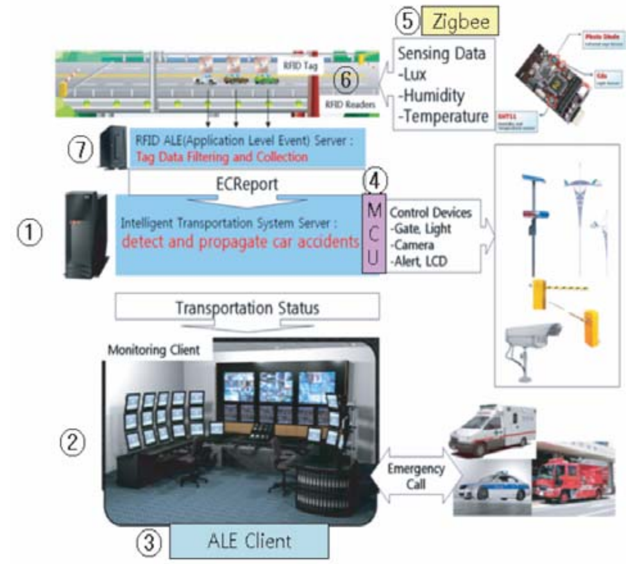

Fig. 2 System architecture

IV. Event Detection Algorithm

The ALE server collects RFID Tag data, and then transmits them to the ITS server as an ECReports format. An ECSpec specifies how an event cycle is to be calculated. The ECSpec are requested using poll or created immediate. The ECReports contain one or more reports generated from poll of an ECSpec [12-15]. The RFID Tag data is stored in Database attached to the ITS server that detects an event in real-time. When a RFID Tag is recognized for the first time, the ITS server begins to track the position of the RFID Tag in real-time. At this time, the position of the RFID tag means the position of a RFID reader which reads the RFID tag. When an event happens, the ITS server reports to an ITS client. Events are categorized into three situations; normal traffic, congestion, and accident.

Fig. 3 shows normal traffic. It means a RFID tag attached to a car has been read by different readers in the direction of the car's movement. The ITS server can ask a query in the database in order to know about the current position of a specific car. Red arrows mean a RFID Reader reads a RFID Tag at a specific space and time.

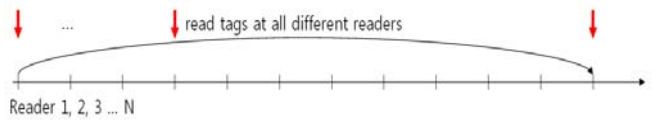

Fig. 3 Normal traffic

Fig. 4 shows an event considered as a car accident occurs at a specific port of a road. The same RFID Tag has been read by the same RFID reader over a certain period of time, but there are no cars in front of the red arrow. This case is considered as a car stop in this paper. We designed our system to propagate accident information to other drivers as quickly as possible to reduce the accident rate and casualties, because it is very dangerous to stop a car on a road. The ITS server can also ask a query in the database in order to propagate information to other drivers.

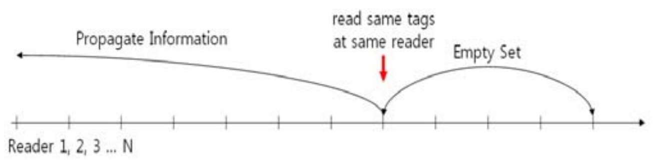

Fig. 4 When an accident happened

Fig. 5 shows congestion caused by a car accident. After an event considered as the first car accident happens, congestion will occur. The ITS server detects congestion then propagates congestion information to other drivers to avoid additional collisions.

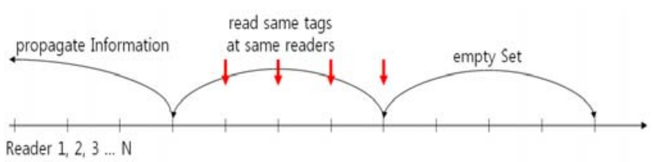

Fig. 5 Congestion

V. Experimental Setup

A. Experimental Setup

Table I summarizes a development environment used in a prototype implementation.

B. A Prototype Implementation

We implemented a small test-bed as shown in Fig. 6. Its length is 2 meters. Lights and alerts were installed on the left and right guardrail and gates were installed on an exit of the bridge. An LCD on the pier gives traffic information to drivers. Ten 13.56 MHz RFID Readers are attached under the bridge.

TABLE I
Development Environment

System	OS	Specification
Server	LINUX Kernel 2.6.x	Intel Core2 1.86 GHz gcc 3.2.2 Compiler MySQL 5.0.27
Client	Windows XP	Visual Studio 2005 (C#)
RFID	TinyOS 1.x	ATmega 128 13.56 MHz
Zigbee	TinyOS 1.x	CC2420 2.4 GHz Base sensors (temperature, illumination, humidity)
MCU Board		Compiler - IAR Embedded workbench ISP downloader : Ponyprog2000

Fig. 6 Bridge model

C. Monitoring Program in ITS Client

The monitoring program consists of Bridge Status, Topology & Statistic, Sensor & Device, Accident History, and Camera Control as shown in Fig. 7. The Bridge Status displays general information such as traffic flow, car accidents, and the state of devices as well as atmospheric phenomena. Topology & Statistic displays statistical weather information near the bridge. Sensor & Device controls devices like sirens, alerts, the LCD and gates, and displays the latest information on weather.

Accident History displays car accident statistics as shown in Fig. 8. Through analysis on occurrence location and occurrence time of the accident, the ITS client reports car accident information to administrators, and helps them to maintain and improve a road. They also give safe driving guidance to drivers. Finally, Camera Control enables administrators to watch the traffic condition on the bridge.

D. Expectable Scenario

The Kim family is going on a picnic by the West Sea. Before arriving at noon, they see the sea and eat shrimp. Accompanied by a sounding alarm, they see the message, "An accident happened. Slow down," while they drive on the Seohae Grand Bridge in a thick fog. They decrease their speed as soon as they get the message. At this time, they see three cars were involved in a car accident 15 meters ahead. However, they avoided the accident thanks to a Safe and Intelligent Bridge System.

Fig. 7 Bridge status

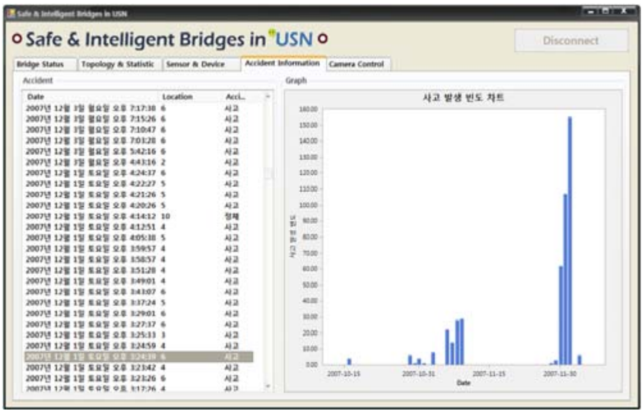

Fig. 8 Accident history

VI. Application Level Event Process Using an EPC

In this section, we explain application level event process with the EPC generator. At first, the ALE server registers logical readers. Next, a logical reader layout is defined as required by the ALE specification. If an event cycle runs, logical readers can read RFID tags. Finally, an ECReports is the output from the event cycle after finishing [13, 14].

A. *Register Logical Readers*

As shown in Fig. 9, the ALE server registers logical readers in advance because it may manage many RFID readers that are used for unrelated purposes. For example, there may be ten sections each having three RFID readers on the bridge; in such a case, a typical ALE request may be directed at the three readers for a particular section, but it is unlikely that an application tracking the flow of cars would want readings from all 30 RFID readers to be combined into a single event cycle. In this paper, the ALE server maps each logical reader to many physical readers using a TCP/IP socket. Each physical reader has an IP address and a port.

B. *Define ALE Specification*

ALE also helps to decouple the data gathering techniques from the physical layout of a business environment through the concept of a logical reader. The logical reader layout is defined as required by the ALE specification, but vendors and developers are allowed to implement in a manner most suitable to their needs. That is, ALE offers flexibility and scalability. If any logical reader is already registered, the ALE client can define an event cycle like the following Fig. 10.

```
- <LogicalReader Name="ExpressHighway15_SeohaeBr_section1">
    <IP>210.115.226.13</IP>
    <Port>9031</Port>
    <IP>210.115.226.13</IP>
    <Port>9032</Port>
    <IP>210.115.226.13</IP>
    <Port>9033</Port>
  </LogicalReader>
- <LogicalReader Name="ExpressHighway15_SeohaeBr_section2">
    <IP>210.115.226.13</IP>
    <Port>9041</Port>
    <IP>210.115.226.13</IP>
    <Port>9042</Port>
    <IP>210.115.226.13</IP>
    <Port>9033</Port>
  </LogicalReader>
</LogicalReaders>
```

Fig. 9 Mapping logical names to physical reader devices

Fig. 10 Procedure to define ECSpec

```
<?xml version="1.0" encoding="UTF-8" ?>
- <ECSpec xmlns:ale="urn:epcglobal:ale:xsd:1" xmlns:epcglobal="urn:epcglobal:xsd:1"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="urn:epcglobal:ale:xsd:1 Ale.xsd"
    shemaVersion="1.0" creationDate="2007-12-09 오후 4:25:47">
  - <LogicalReaders>
      <logicalReader>ExpressHighway15_SeohaeBr_section1</logicalReader>
    </LogicalReaders>
  - <BoundarySpec>
      <startTrigger>ALE1</startTrigger>
      <repeatPeriod unit="MS">1000</repeatPeriod>
      <stopTrigger>ALE2</stopTrigger>
      <duration unit="MS">1000</duration>
    </BoundarySpec>
  - <reportSpecs>
    - <reportSpec reportName="TrafficInformation">
        <reportSet set="CURRENT" />
        <output includeTag="Yes" />
      </reportSpec>
    </reportSpecs>
</ECSpec>
```

Fig. 11 ECSpec Document

The event cycle named as "ExpreeHighway15_Seohea_section1" is chosen. After the ALE client defines ECBoundarySpec which specifies how the beginning and end of event cycles are to be determined, it generates "TrafficInformation" as a report. Basically, only the RFID Tag in the report is transmitted (duration 1000ms, period 1000ms). An ECSpec is posted as an XML document using a define method as shown in Fig. 11. Fundamentally, ALE recommends using XML documents for the communication of the heterogeneous systems. The ALE server starts to collect RFID Tag data when the specified startTrigger is received.

As shown in Table II below, an ECSpec contains a list of logical readers, a specification of how the boundaries of event cycles are to be determined, and a list of specification each of which describes a report to be generated from this event cycle.

TABLE II
ECSPEC

readers : List
boundaries : ECBoundaySpec
reportSpecs : List

As shown in Table III, an ECBoundarySpec specifies how the beginning and end of event cycles are to be determined. The startTrigger and stopTrigger parameters are optional, and the startTrigger and repeated parameters is mutually exclusive. Once an event cycle starts, it extends until one of the following is true. The specified duration expires, the specified stopTrigger is received, or the ECSpec changes to the unrequested state.

TABLE III
ECBOUNDARYSPEC

startTrigger : ECTrigger
repeatPeriod : ECTime
stopTrigger : ECTrigger
duration : ECTime

As shown in Table IV, an ECReportSpec specifies one report to be returned for executing an event cycle. An ECSpec contains a list of one or more ECReportSpec instances. It consists of ECReportSetSpec, ECFilterSpec, ECGroupSpec, and ECReportOutputSpec. The ECReportSetSpec specifies what set of EPCs is considered for reporting, the ECFilterSpec specifies how the raw EPCs are filtered before inclusion in the report, the ECGroupSpec specifies how the filtered EPCs are grouped together for reporting, and the ECReportOutputSpec specifies whether to return the EPC groups themselves or a count of each group, or both.

TABLE IV
ECREPORTSPEC

reportName : string
reportSet : ECReportSetSpec
filter : ECFilterSpec
group : ECGrouptSpec
output : ECReportOutputSpec
reportIfEmpty : boolean
reportOnlyOnChange : Boolean

C. *Generate EPC*

If an event cycle runs, logical readers can read RFID tags. However because our test-bed work is on a small-scale, we developed a simulator to generate EPCs and forward them to the ALE server through logical reader's IP and port as shown in Fig. 12 below. It should be mapped into registered physical readers exactly.

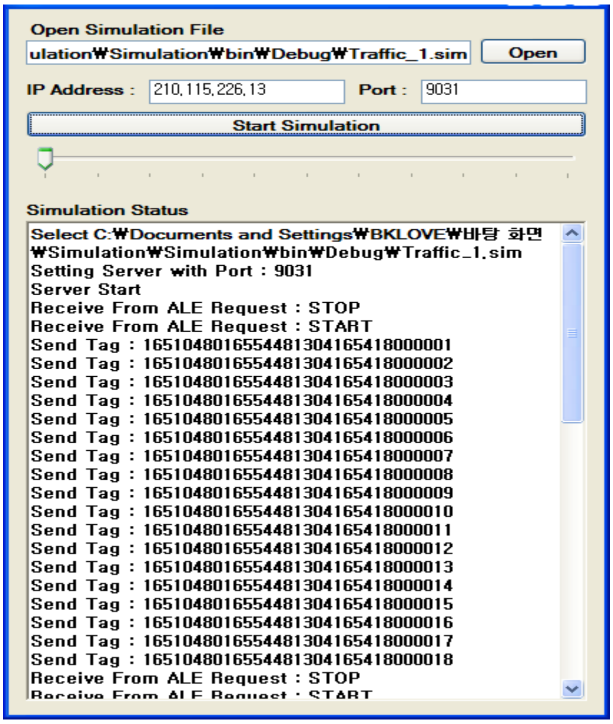

Fig. 12 EPC generator

Assuming that all cars are moving toward a road, their mobility model is shown in Table V below [15].

TABLE VI
ECREPORTS

specName : string
date : dateTime
ALEID : string
totalMilliseconds : long
terminationCondition : ECTerminationCondition
spec : ECSpec
reports : List

D. *Report ECReport*

An ECReports is the output from the event cycle "ALE_IN_ITSServer" as shown in Fig. 13.

As shown in Table VI, the specName is the name of the

ECSpec that controlled this event cycle, the data represents the data time when the event cycle ended, the ALE ID is an identifier for the deployed instance of ALE implementation, the totalMilliseconds is the total time in milliseconds, the terminationCondition indicates what kind of event caused the event cycle to terminate, and the spec is a copy of ECSpec that generated this ECReports instance.


```
<?xml version="1.0" encoding="UTF-8" ?>
- <ale:ECReports xmlns:ale="urn:epcglobal:ale:xsd:1" xmlns:epcglobal="urn:epcglobal:xsd:1" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
    instance" xsi:schemaLocation="urn:epcglobal:ale:xsd:1 Ale.xsd" schemaVersion="1.0" creationDate="2007-12-10T10:54:06.444-05:00"
    specName="ALE_IN_ITSServer" date="2007-12-10 오전 1:36:050" ALEID="ALE1" totalMilliseconds="1000"
    terminationCondition="DURATION">
  - <reports>
    - <report reportName="TrafficInformation">
      - <member logicalReader="ExpressHighway15_SeohaeBr_section1" date="2007-12-10 오전 1:36:05">
          <epc>urn:epc:tag:gid-96:1651048016554481304165418000001</epc>
        </member>
      - <member logicalReader="ExpressHighway15_SeohaeBr_section1" date="2007-12-10 오전 1:36:06">
          <epc>urn:epc:tag:gid-96:1651048016554481304165418000002</epc>
        </member>
      - <member logicalReader="ExpressHighway15_SeohaeBr_section1" date="2007-12-10 오전 1:36:06">
          <epc>urn:epc:tag:gid-96:1651048016554481304165418000003</epc>
        </member>
      - <member logicalReader="ExpressHighway15_SeohaeBr_section1" date="2007-12-10 오전 1:36:06">
          <epc>urn:epc:tag:gid-96:1651048016554481304165418000004</epc>
        </member>
      - <member logicalReader="ExpressHighway15_SeohaeBr_section1" date="2007-12-10 오전 1:36:06">
          <epc>urn:epc:tag:gid-96:1651048016554481304165418000005</epc>
        </member>
      - <member logicalReader="ExpressHighway15_SeohaeBr_section1" date="2007-12-10 오전 1:36:06">
          <epc>urn:epc:tag:gid-96:1651048016554481304165418000006</epc>
        </member>
      - <member logicalReader="ExpressHighway15_SeohaeBr_section1" date="2007-12-10 오전 1:36:06">
          <epc>urn:epc:tag:gid-96:1651048016554481304165418000007</epc>
        </member>
      - <member logicalReader="ExpressHighway15_SeohaeBr_section1" date="2007-12-10 오전 1:36:06">
          <epc>urn:epc:tag:gid-96:1651048016554481304165418000008</epc>
        </member>
      - <member logicalReader="ExpressHighway15_SeohaeBr_section1" date="2007-12-10 오전 1:36:06">
```


Fig. 13 ECReport Document

TABLE VI
ECREPORTS

specName : string
date : dateTime
ALEID : string
totalMilliseconds : long
terminationCondition : ECTerminationCondition
spec : ECSpec
reports : List

VII. EVENT DETECTION AND REACTION ITS SERVER

In this section, we show the precedures for Event Detection and Reaction in ITS server. Normally, a car passes through the bridge safely. However, it could have broken down due to car accidents or mechanical failures. At this time, RFID readers installed on the bridge reads RFID Tags attached to cars moving on the bridge. After the ALE server collects and filters EPCs read by the registered logical readers, it creates ECReports. The ITS server inputs the ECReports in the database. Such the ECReports could be used for tracking a car or detecting a car accident.

A. Queries for Event Detection

The ITS server classifies the ECReport into three fields such as RFIDTagID, Logical Reader, and Register Time, and then it stores the ECReport in Database using the following query Q1.

Q1. Input ECReports in the Database

INSERT INTO READ_TAGDATA (LogicalReader, RFIDTagID, CurrTime)
VALUES ([Text], [Text], [TimeValue]);

ITS server can also analyze vehicular traffic flow using the following query Q2.

Q2. Find the location of RFIDTags

SELECT RFIDTagID, LogicalReader, CurrTime **AS** TI
FROM READ_TAGDATA
WHERE CurrTime >= [TimeValue]
GROUP BY RFIDTag **HAVING** MAX(CurrTime);

Fig. 14 shows the results of the query Q1.

RFIDTagID	Logical Reader	Register Time
203854761E226E81	ExpressHighway15_SeohaeBr_Section1	2007-12-10 오전 1:36:05
203854761E226E82	ExpressHighway15_SeohaeBr_Section1	2007-12-10 오전 1:36:06
203854761E226E83	ExpressHighway15_SeohaeBr_Section1	2007-12-10 오전 1:36:06
203854761E226E85	ExpressHighway15_SeohaeBr_Section1	2007-12-10 오전 1:36:06
203854761E226E86	ExpressHighway15_SeohaeBr_Section1	2007-12-10 오전 1:36:06
203854761E226E87	ExpressHighway15_SeohaeBr_Section1	2007-12-10 오전 1:36:06
203854761E226E88	ExpressHighway15_SeohaeBr_Section1	2007-12-10 오전 1:36:06

Fig. 14 The query Q1 result on RFID TagECReports input

Fig. 15 shows the result of the query Q2. For tracking a specific car, the ITS server finds out the RFID Tag which is attached to the car. After looking up the latest updated RFID tag in the table below, its logical reader indicates its current position.

RFIDTagID	Logical Reader	Register Time
203854761E226E81	ExpressHighway15_SeohaeBr_Section1	2007-12-10 오전 1:36:05
203854761E226E82	ExpressHighway15_SeohaeBr_Section1	2007-12-10 오전 1:36:06
203854761E226E83	ExpressHighway15_SeohaeBr_Section1	2007-12-10 오전 1:36:06
203854761E226E84	ExpressHighway15_SeohaeBr_Section1	2007-12-10 오전 1:36:06
203854761E226E81	ExpressHighway15_SeohaeBr_Section2	2007-12-10 오전 1:36:07
203854761E226E83	ExpressHighway15_SeohaeBr_Section2	2007-12-10 오전 1:36:08

Fig. 15 The query Q2 result on RFID Tag Location

Finally, the ITS server is able to detect a car accident on a specfic location using the following query Q3.

Q3. Detect a car accident event.

SELECT R, T, COUNT(*) **AS** CNT
FROM (**SELECT** ReaderID AS R, RFIDTag **AS** T, CurrTime
FROM READ_TAGDATA
WHERE CurrTime <= T1 **AND** CurrTime >= T2)
GROUP BY R, T;

Fig. 16 shows the result of the query Q3 on detecting the accident. A specific RFID tag is read more than 5 times on ExpressHighway15_SehaeBr_section3 which means the car with it stopped on the road. It may be considered as a car accident.

RFIDTagID	Logical Reader	Count
203854761E226E81	ExpressHighway15_SeohaeBr_Section1	1
203854761E226E81	ExpressHighway15_SeohaeBr_Section2	1
203854761E226E81	ExpressHighway15_SeohaeBr_Section3	1
203854761E226E84	ExpressHighway15_SeohaeBr_Section1	1
203854761E226E84	ExpressHighway15_SeohaeBr_Section2	1
203854761E226E83	ExpressHighway15_SeohaeBr_Section3	5

Fig. 16 The query Q3 result on a car accident

B. Reaction to Event Detection

When the ITS server detects a car accident, it sends signals to the MCU (Micro Control Unit) which controls sirens, alerts, LCD and gates. By propagating accident information to other drivers quickly, it can prevent additional collisions. Control signals are transmitted through UART and the state of each device is stored in Database. If drivers have a Zigbee device, it has them know accident information in the car. Accident history is available to analyze the accident situation by recording occurrence location and occurrence time of the accident, and atmospheric information such as temperature, illumination and humidity. On the whole, the ITS server helps administrators to decide whether an event is a car accident or not by watching the scene of the expected accident through a camera.

VIII. Conclusion

In conclusion, we propose Safe & Intelligent Bridges System based on ALE-compliant RFID middleware in USN. This system was designed to detect an event such as a car crash in real-time and propagate accident information quickly. We also implemented a small test-bed and developed a simulator which generates RFID tags considering a car’s movement. The ALE-compliant RFID middle can discriminate among different events such as normal traffic condition, congestion, and accidents. Its interface provides independence between infrastructure components that acquire the raw EPC data, architectural components that filter and count the data, and applications that use the data. Specially, it offers flexibility and scalability to systems. Our system proposed in this paper provides a novel way to manage traffic systems safely and efficiently by complementing problems in the current traffic system.

Acknowledgment

This research was supported by KOTEF (Korea Industrial Technology Foundation).

References

[1] Road Traffic Safety Authority, http://www.rtsa.co.kr
[2] Ministry of Construction & Transportation, http://www.moct.go.kr
[3] Lee, H., “A Study on ITS in Domestic and Advanced Countries,” ETRI Journal, Vol. 14, No. 2, 1999
[4] TranSystems corp., “Statewide Transit Intelligent Transportation Systems Deployment Plan,” Iowa Department of Transportation, May 2002.
[5] Reynold, Stuart, “Architectural Trends: A real-time distributed database for Europe,” Traffic Technology International, Feb/Mar 1998
[6] WinRFID – A Middleware for the enablement of Radio Frequency Identification (RFID) based Applications, B.S. Prabhu, X. Su, H. Ramamurthy, C-C, R. Graph. In Mobile, Wireless and Sensor Networks: Technology, Application and Future Directions, Eds. Rajeev Shorey, et al., John Wiley, 2006, 313-338.
[7] RFID Handbook: Fundamentals and Applications in Contactless Smart Cards and Identification, 2ed, Klaus Finkenzeller, Wiley, 2003
[8] Auto-ID Labs, http://www.autoidlabs.org/
[9] Mimbela L. Y., Klein L.A., “A summary of vehicle detection and surveillance technologies used in intelligent transportation systems,” New Mexico State University, Tech. Rep., November. 8.
[10] Xiao-jun Tan, Jun Li, and Chunlu Liu, "A video-based real-time vehicle detection method by classified background learning," World Transactions on Engineering and Technology Education 2007 UICEE Vol.6, No.1, 2007
[11] Chain-reaction collision, http://joongangdaily.joins.com
[12] EPCglobal, The EPCglobal architecture framework, final version, July 1, 2005.
[13] EPCglobal. EPC Information Services (EPCIS) Version 1.0 Specification, working draft, June 8, 2006.
[14] K. Traub, S. Bent, T. Osinski, S. N. Perertz, S. Rehlinh, S. Rosenthat, and B. Tracey, The Application Level Event(ALE) Specification, ver1.0, 2005.
[15] Zhiwei Cen, et al., “Motion Prediction in Mobile Communication Systems from a Historical and Geographical and Perspective,” CSE812 Group Report, Fall 2002.