J. Internet Comput. Serv.
ISSN 1598-0170 (Print) / ISSN 2287-1136 (Online)
http://www.jics.or.kr

하이퍼레저 패브릭을 이용한 화물차 디지털 운행기록 단말기의 안전운행 보상시스템 구현☆

Development of The Safe Driving Reward System for Truck Digital Tachograph using Hyperledger Fabric

김 용 배[1] 백 주 용[2] 김 종 원[3*]
Yong-bae Kim Juyong Back Jongweon Kim

요 약

본 논문의 안전운행 보상시스템은 안전운전을 수행한 차량운전자에게 직접적인 보상을 제공하여 안전운전의 동기를 부여하고 적극적 참여를 유도함으로써 사고의 발생을 줄여 생명과 재산의 손실을 줄이는데 목표가 있다. 기존의 디지털 운행기록계의 경우 차량의 운전상태를 기록만 하였으나, 안전운전보상시스템은 사고예방 효과를 높이기 위한 지원책으로서 안전운전을 수행한 경우 금전적 보상을 통해 위험운전을 피하고 안전운전을 하도록 유도하였다. 즉, 과속으로 인한 사고 발생 빈도가 높은 지역에서는 속도 준수, 또는 차 간 거리 유지, 지정차로 운행 등의 안전운행 지시를 수행한 경우 직접적인 보상을 제공함으로써 안전운전 동기를 부여하여 교통사고를 예방하고자 한다. 이러한 안전운행 데이터와 보상내용은 투명하고 안전하게 관리되어야 하므로 보상근거와 보상내용을 폐쇄형 블록체인 하이퍼레저 패브릭을 이용하여 구축하였다. 그러나 블록체인 시스템은 투명성과 안전성이 보장되는 반면에 낮은 데이터 처리속도가 문제가 되므로 이를 개선하고자 블록생성 가속 기능을 구현하였다. 본 연구에서는 순차적으로 블록을 생성하는 속도가 10TPS(Transaction per second) 내외의 낮은 속도를 나타내어, 블록의 생성속도를 높이기 위해 가속 기능을 적용한 결과 1,000TPS 이상의 고성능 네트워크를 구현하였다.

☞ 주제어 : 블록체인, 디지털운행기록계, 하이퍼레저패브릭, 경량블록체인, 랜덤시드, 사고안전보상시스템, 안전채널, 보상채널

ABSTRACT

The safe driving reward system aims to reduce the loss of life and property by reducing the occurrence of accidents by motivating safe driving and encouraging active participation by providing direct reward to vehicle drivers who have performed safe driving. In the case of the existing digital tachograph, the goal is to limit dangerous driving by recording the driving status of the vehicle whereas the safe driving reward system is a support measure to increase the effect of accident prevention and induces safe driving with financial reward when safe driving is performed. In other words, in an area where accidents due to speeding are high, direct reward is provided to motivate safe driving to prevent traffic accidents when safe driving instructions such as speed compliance, maintaining distance between vehicles, and driving in designated lanes are performed. Since these safe operation data and reward histories must be managed transparently and safely, the reward evidences and histories were constructed using the closed blockchain Hyperledger Fabric. However, while transparency and safety are guaranteed in the blockchain system, low data processing speed is a problem. In this study, the sequential block generation speed was as low as 10 TPS(transaction per second), and as a result of applying the acceleration function a high-performance network of 1,000 TPS or more was implemented.

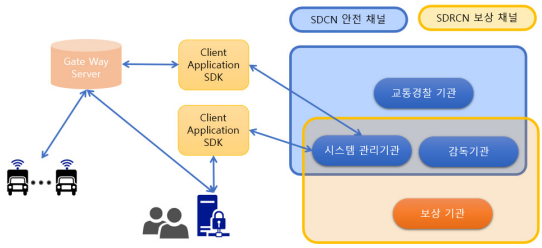
☞ keyword : Blockchain, DTG(Digital Tacho-Graph), Hyperledger Fabric, Lightweight Blockchain, Random Seed, SDRS(Safe Driving Reward System), SDCN(Safe Driving Channel Network), SDRCN(Safe Driving Reward Channel Network)

1. 서 론

기존의 디지털운행기록계(DTG: Digital Tacho Graph)의 설치는 사업용 차량이 자가용보다 사고율이 5배 이상 높고, 교통법규 위반 건수가 1.7배 높아 난폭운전 개선 목적으로 교통안전법에 따라 2011년 1월 1일 이후 신규로 등록하는 사업용 차량에 운전상태를 나타내는 속도와

1 Industry-University Cooperation Foundation, Sangmyung University, Seoul, 03016, Korea.
2 Quantum Gate Inc., Sejong, 30043, Korea
3 Dept. of IIOT, Sangmyung University, Seoul, 03016, Korea.
* Corresponding author (jwkim@smu.ac.kr)
[Received 5 February 2022, Reviewed 16 March 2022(R2 30 March 2022), Accepted 16 April 2022]
☆ This work was granted by Ministry of Land, Infrastructure and Transport of Korea, 20CTAP-C152294-02.

RPM, 브레이크, GPS를 통한 위치, 방위각, 가속도, 주행거리 등을 자동으로 전자식 기억장치에 기록하도록 의무화되어 운행되고 있다[1]. 또한, 어린이 탑승차량의 사고가 빈번해짐에 따라 어린이집, 유치원, 학원 등의 어린이 보호 차량으로 확대 시행하도록 2020년 4월 29일 교통안전법 개정안[2]이 의결되었다. 디지털운행기록계의 운행기록은 교통안전관리공단에서 주기적으로 수집하여 안전운전에 관한 분석을 수행하도록 하고 있으며, 분석결과는 자동차·운전자·교통수단 운영자에 대한 운행관리, 교육. 훈련, 운행경로 개선과 교통사고 예방을 위한 교통정책 수립에 활용되고 있다. 본 연구에서 제안하는 사고안전 보상 시스템(SDRS: Safe Driving Reward System)은 그림1과 같이 사고안전 채널 네트워크(SDCN: Safe Driving Channel Network, 안전채널)와 사고안전 보상 채널 네트워크(SDRCN: Safe Driving Reward Channel Network, 보상채널)로 구성되어 있다.



(그림 1) 블록체인기반 안전운전 보상시스템 구성도
(Figure 1) Safe Driving Reward System Configuration

2. 연구동향

2.1 블록체인

Blockchain은 chain algorithm을 적용한 기술로써 다수의 데이터를 암호화하여 chain으로 연결함으로써 위·변조를 방지하고 탈중앙 분산 저장 방식으로 데이터를 관리하는 알고리즘이다. 체인으로 연결된 데이터 구조는 linked list 방식과 유사하여 hash 자체가 각 노드를 연결하는 주소 역할을 한다.

블록체인의 본격적인 연구는 Satoshi Nakamoto[3]가 비트코인 암호화폐 개발을 제안하고 2009년 1월 블록체인 알고리즘 기반의 비트코인을 개발한 것이다. 블록체인이란 다수의 거래내역을 묶어 하나의 블록을 구성하고, 해시를 이용하여 여러 블록을 chain 형태로 연결하여 구성한다. 블록체인으로 구성된 데이터는 여러 곳에 복사본을 저장하는 알고리즘이다. 이러한 특징으로 블록체인은 위·변조가 불가능하고 탈중앙 분산 관리가 가능하게 되었으며, 비트코인을 시작으로 Litecoin[4], Ethereum [5], Ripple[6], bitcoin cash[7], EOS[8] 등 다양한 암호화폐에 적용되었다.

블록체인은 여러 가지 장점에도 불구하고 느린 처리속도가 걸림돌이다. 비트코인의 경우 하나의 새로운 블록을 구성하려면 약 10분, Litecoin 5분, Ethereum 15초 정도 소요되며, 거래속도는 비트코인이 7TPS, Litecoin 56TPS, Ethereum 20TPS, 정도이다[9].

블록체인 시스템은 누구나 참여할 수 있는 형태의 개방형 블록체인(Public Blockchain)과 참여자를 엄격히 제한하는 폐쇄형 블록체인(Private Blockchain)으로 나눌 수 있다. 개방형 블록체인의 특징은 채굴이라는 과정을 통하여 블록체인 시스템을 운영하고 있으며, 폐쇄형 블록체인 시스템은 관리기관의 책임하에 시스템을 구축하고 참여자를 관리한다. 이러한 특징을 비교하여 표 1에 나타내었다[10].

(표 1) 블록체인 시스템 비교
(Table 1) Comparison of Blockchain Networks

구분	공개형 블록체인	폐쇄형 블록체인
조직참여	조직의 승인 불필요	조직의 승인 필요
보상	암호화폐 발행으로 보상, (transaction fee)	필요 없거나, 거의 없음
장점	높은 안전성과 신뢰성 투명성, 익명성	높은 효율성과 확장성
단점	블록생성속도 늦음	신뢰시스템으로 보안성 취약
합의	POW, POS, DPOS	BFT, RAFT, Solo, Kafka
사용례	비트코인, 이더리움, 모네로 등	리플, 하이퍼레저 패브릭 등

2.2 경량 블록체인(Lightweight Blockchain)

블록체인 기술은 암호화 기술을 이용하여 해킹할 수 없는 신뢰성을 제공한다. 그러나 블록체인 기술의 낮은 데이터처리율은 시스템 적용에 단점으로 작용하여 차량과 같은 이동형 단말기의 적용에 어려움이 있다. 이러한 단점을 극복하기 위하여 개발된 기술이 경량 블록체인이다. 경량 블록체인은 보안성을 유지하면서 IoT(Internet of Things) 단말기나 자율주행 차량 등에 적용되는 단순화된

블록체인 알고리즘이다[11]. 본 연구에서는 화물차의 특성상 온라인, 오프라인 환경에서 모두 사용할 수 있는 시스템이 요구되었으며 차량단말기와 게이트웨이 서버로 경량 블록체인 시스템을 구성하였다. 표 2에는 본 연구에서 사용한 경량 블록체인 시스템과 폐쇄형 블록체인인 하이퍼레저 패브릭 블록체인의 특징을 비교하였다.

(표 2) 폐쇄형 블록체인과 경량 블록체인 비교
(Table 2) Private Blockchain vs Lightweight Blockchain

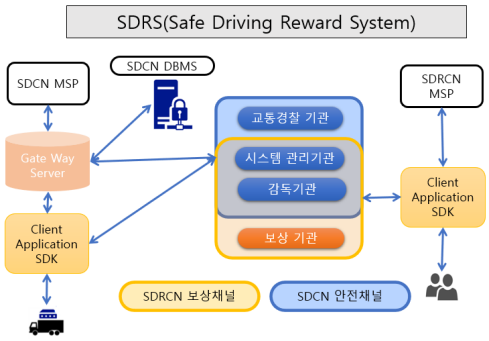
구 분	Private Blockchain	Lightweight Blockchain
합의알고리즘	Raft, Solo etc	없음
서명	CA (Certificate Authority)	서버에서 등록한 key 값
블록생성	오더러(orderer)	단말기에 블록 생성
인증방법	MSP를 이용한 전자지갑	암호 로그인
실행방법	온라인 상태	온라인, 오프라인

3. 연구내용

3.1 사고안전 보상시스템(SDRS)의 개요

사고안전 보상시스템은 하이퍼레저패브릭 기반의 블록체인 네트워크이다. 하이퍼레저패브릭 블록체인 네트워크에서는 채널단위로 원장을 공유하게 되는데 같은 채널에 포함된 조직과 참여자는 원장의 데이터 읽기, 쓰기 등의 접근권한을 관리할 수 있다.

사고안전보상시스템은 두 개의 채널로 구성되어 있으며, 하나의 채널에만 소속되어 있는 기관이 있어 다른 채널의 원장에 접근할 수 없으며, 두 개의 채널 모두에 소속되어 있는 기관은 양쪽 채널의 데이터에 접근할 수 있다. 그림 2와 같이 안전채널에서는 차량의 운행정보와 안전운전 행위에 대한 데이터를 안전채널에 기록 관리하며, 보상채널은 안전채널 정보를 바탕으로 안전운전 행위에 대한 금전적 보상정보를 장부에 기록 관리하고 있다.



(그림 2) 블록체인 기반 사고안전 보상시스템 구성
(Figure 2) Safe Driving Reward System Configuration

(표 3) 안전채널(SDCN)과 보상채널(SDRCN)
(Table 3) SDCN Channel vs SDRCN Channel

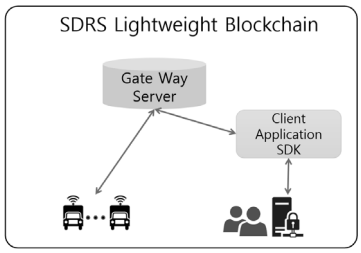
참여기관	SDCN 안전 채널	SDRCN 보상 채널
운영기관	○	○
감독기관	○	○
교통경찰	○	
보상기관		○

표 3과 같이 안전채널은 운영기관, 감독기관, 교통경찰이 시스템운영에 참여하며, 보상채널은 운영기관, 감독기관, 보상기관 등이 참여할 수 있다.

운영기관은 시스템 유지보수를 담당하며, 감독기관은 사고안전 보상시스템이 정상적으로 운용되는지를 관리감독하고, 데이터의 입출력을 담당하기 위하여 안전채널 및 보상채널에 모두 접근할 수 있다. 교통경찰의 경우 화물차의 안전운전을 지원할 수 있도록 안전채널의 정보를 읽을 수 있는 권한을 가지며, 보상기관은 보상에 관련된 참여자의 보상채널 데이터에 접근을 관리하기 위한 기관이다.

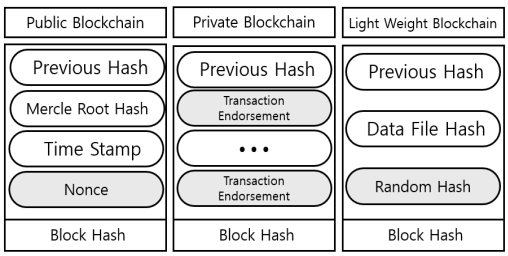
3.2 경량 블록체인 시스템 적용

본 연구에서 개발된 경량 블록체인 시스템은 온, 오프라인 모두에서 동작할 수 있는 특징을 가지고 있다. 차량운행 중 네트워크 접속이 단절된 경우 서버와 단말기 간의 상호인증과정을 수행할 수 없게 된다. 이 경우 서버에서 제공한 랜덤시드(Random Seed)를 활용하여 오프라인에서도 안전하게 사용할 수 있는 알고리즘을 개발 적용하였다[12].



(그림 3) 경량 블록체인 네트워크
(Figure 3) Lightweight Blockchain Network

랜덤시드를 활용한 경량 블록체인 시스템은 폐쇄형 블록체인 시스템이나 개방형 블록체인 시스템과 블록헤더의 구조에서 차이를 보인다.



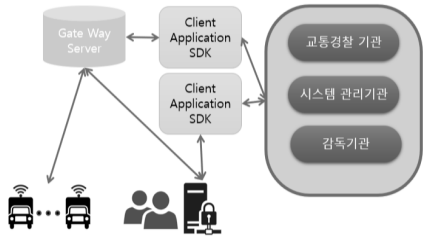
(그림 4) 블록체인 구조의 비교
(Figure 4) Comparison of Blockchain Architectures

개방형 블록체인의 블록구조는 그림 4[12]와 같이 Nonce를 이용한 작업증명으로 위변조방지 기능을 수행하며, 폐쇄형 블록체인 시스템은 합의 과정에 참여한 피어노드(Peer Node)의 서명을 이용하여 위변조 방지기능을 수행한다. 경량 블록체인에서는 서버에서 제공된 랜덤시드를 이용하여 통신이 끊긴 상태에서도 위변조 방지가 가능하게 한다.

3.3 안전채널(SDCN)

3.3.1 안전채널(SDCN)의 개요

안전채널은 운행 중 위험 구간이나 눈, 비, 안개 등의 상황에서 안전운전을 수행한 보상을 제공할 목적으로 마일리지 포인트를 제공하거나 과속, 차간거리 미확보 등 난폭운전을 한 경우 이에 대한 불이익을 주어 교통사고를 예방하고 안전운전을 습관화할 수 있도록 지원하는 시스템이다. 그림 5와 같이 차량에 설치된 단말기에는 온라인 또는 오프라인 상태에서 안전운전에 관련된 데이터를 저장하고 위변조 방지를 위한 경량 블록체인이 적용된다.



(그림 5) 안전채널 네트워크
(Figure 5) Safe Driving Channel Network

단말기에 기록된 데이터는 게이트웨이를 통하여 SDCN 데이터베이스에 등록된다. 데이터베이스에 사고안전 데이터가 등록되면 분산응용프로그램(DAPP)을 통하여 안전채널 블록체인에 데이터 셋을 기록하게 된다. 또한, 관리자나 안전채널에 등록된 사용자는 응용프로그램을 통해 체인코드를 실행하여 블록체인 장부를 검색하거나 변경할 수 있다.

3.3.2 안전채널(SDCN) 데이터 셋

안전채널의 데이터 셋은 차량이 운전 중에 발생되는 안전운전에 관련된 이벤트를 기록한다. 하이퍼레저 패브릭 블록체인의 데이터기록 방법은 키(key)와 값(Value)의 쌍을 갖는 상태 데이터베이스(World State Database)에 키와 값을 기록 변경하여 현 상태 값을 유지하도록 구성되어 있다. 상태 데이터베이스는 현재의 상태만을 기록하고 중복된 데이터가 입력되면 이전의 상태는 블록에 기록하고 변경된 값만 상태 데이터베이스에 기록한다.

하이퍼레저 패브릭에서 체인코드(Smart Contract)는 블록체인에 데이터를 입출력할 수 있는 인터페이스를 제공하며, 데이터 셋에 접속 가능한 체인코드를 개발하여 피어노드(Peer Node)에 설치하여야 한다. 표 4는 안전채널에서 사용하는 데이터 셋의 구성이다.

dtgNo는 일련번호로 키를 나타내며, eventDate (언제), GPSx, GPSy(어디서), eventID(어떤) 이벤트가 발생했는지의 정보를 나타낸다. pointValue (마일리지) 값은 안전운전에 도움이 되는 이벤트는 양의 값, 위험운전에 해당하면 음의 값을 주며, 마일리지를 보상금액으로 환산하기 위한 변환율 cRate를 사용하였다.

(표 4) 안전채널 데이터셋
(Table 4) Data Set of the SDCN

항목	내용
dtgNo	일련번호
driverID	운전자 번호
GPSx	위치정보 위도
GPSy	위치정보 경도
roadID	도로번호
eventID	이벤트 코드
pointValue	마일리지 포인트
cRate	환산값
eventDate	이벤트 발생 시간

블록체인 네트워크에 체인코드가 설치되면 명령입력 인터페이스(CLI:Command Line Interface)를 통하여 체인코드를 호출하거나, 분산응용프로그램(DAPP)을 개발하여 체인코드를 호출한다. 본 연구에서는 분산응용프로그램을 개발하여 적용하였다.

3.3.3 안전채널(SDCN) 분산응용프로그램(DAPP)

체인코드를 호출하기 위한 분산응용프로그램의 함수들과 기능은 표 5와 같다. 함수명칭의 명명 방법으로는 T로 시작하는 함수는 순차 블록처리 능력을 측정하는 측정함수이며, Dtg는 안전채널에서 사용되는 함수를 나타낸다. Create, Read, Update, Delete 명칭은 블록의 생성, 읽기, 수정, 삭제를 의미한다. 접두어로 사용된 Ccm은 병렬처리 함수를 나타내며, 접미어로 Asset은 블록체인이 운용 중에 사용하는 함수이다. History는 특정블록의 변화 과정 즉, 생성, 변경, 삭제의 모든 과정을 검색하는 함수이며, ByDriverID는 특정운전자에 해당하는 블록을 호출하는 함수이다.

TDtgCreate는 순차 블록생성 성능측정 함수로 체인코드를 순차적으로 호출하여 실행될 때마다 하나의 블록을 생성하고 블록이 완성되면 다음 블록을 생성하도록 하는 성능측정 함수다.

순차블록 생성의 특성상 블록당 한 개의 메시지만 담고 있어 블록처리 성능이 낮은 결과를 나타낸다. TDtgRead, TDtgUpdate, TDtgDele 함수는 순차읽기, 순차수정, 순차삭제를 수행하여 하나의 블록을 처리한 후 다음 블록을 순차적으로 처리하여 성능을 측정하는 함수다. CcmDtgCreate, CcmDtgRead, CcmDtgUpdate, CcmDtgDelete 함수는 이름에서 알 수 있듯이 병렬처리로 블록을 처리함으로써 성능을 향상시킨 함수다. DtgHistoryAsset 함수는 key값에 해당하는 모든 블록을 반환하는 함수이다. 즉 블록의 모든 변경이력이 출력된다. 만약 블록이 삭제된 경우 블록의 모든 필드값은 리셋되며, 블록삭제 상태를 나타내는 논리값 FALSE(0)를 출력한다. 운전자별 블록정보 읽기 함수 DtgReadByDriverIDAsset은 상태 데이터베이스에서 운전자와 관련된 데이터를 출력하는 함수로 운전자의 정보 검색에 사용된다.

(표 5) 안전채널 분산응용 프로그램의 함수
(Table 5) DAPP function list of SDCN

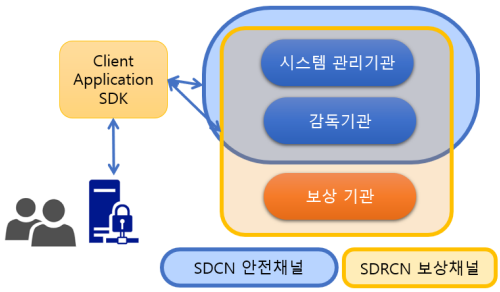
함수명칭	기능
TDtgCreate	순차 블록생성 성능측정함수
TDtgRead	순차 블록읽기 성능측정함수
TDtgUpdate	순차 블록수정 성능측정함수
TDtgDelete	순차 블록삭제 성능측정함수
CcmDtgCreate	병렬처리 블록생성 성능측정함수
CcmDtgRead	병렬처리 블록읽기 성능측정함수
CcmDtgUpdate	병렬처리 블록수정 성능측정함수
CcmDtgDelete	병렬처리 블록삭제 성능측정함수
CcmDtgCreateAsset	병렬처리 블록생성 함수
CcmDtgReadAsset	병렬처리 블록읽기 함수
CcmDtgUpdateAsset	병렬처리 블록수정 함수
CcmDtgDeleteAsset	병렬처리 블록삭제 함수
DtgHistoryAsset	킷값에 해당하는 모든 블록정보 읽기 함수
DtgReadByDriverIDAsset	운전자별 블록정보 읽기 함수

3.4 보상채널(SDRCN)

3.4.1 보상채널(SDRCN)의 개요

보상채널은 안전채널에 기록된 마일리지 포인트를 이용하여 보상을 제공한다. 따라서 이 정보를 바탕으로 차량을 운행한 운전자에게 보상이 주어지며, 주어진 보상은 운전자별로 포인트가 누적된다.

관리기관의 경우 보상채널과 안전채널의 데이터에 접근할 수 있으며 안전채널에 기록된 장부를 열람하여 운전자별 보상 포인트를 보상 네트워크 장부에 기록하도록 한다. 기록된 보상 포인트는 증감 정보를 장부에 기록하도록 구성되었다.



(그림 6) 보상채널 네트워크
(Figure 6) Safe Driving Reward Channel Network

3.4.2 보상채널(SDRCN) 데이터 셋

보상채널의 데이터 셋은 차량 운행 중에 발생하는 안전운전에 관련된 이벤트가 기록된 안전채널의 정보를 읽고 보상 포인트로 환산하여 보상채널에 기록되는 값이다. 보상 포인트 값은 증감 값으로 표현되며, 안전운전 이벤트의 경우는 양의 값이 되고, 위험운전 이벤트의 경우 음의 값이 된다. 이때 사용되는 보상 네트워크의 데이터 셋은 표 6과 같다.

(표 6) 보상채널 데이터 셋
(Table 6) Data Set of the SDRCN

항목	내용
dtgNo	일련번호
driverID	운전자 번호
eventID	이벤트 코드
eventDate	이벤트 발생 일시
dtgValue	보상 포인트 증감값

보상채널에 참여한 기관이나 사용자는 데이터 셋에 접근할 수 있는 분산응용프로그램의 함수를 이용하여 블록을 생성, 변경, 검색, 삭제를 수행한다.

3.4.3 보상채널(SDRCN) 분산응용프로그램

체인코드를 호출하기 위한 분산응용프로그램의 함수들과 기능은 표 7과 같다.

안전채널 함수에 추가된 것은 보상 잔액을 검색하는 CcmBanlanceQueryAsset 함수와 운전자별 보상 잔액의 변동 내역을 보여주는 CcmBanlanceHistoryAsset 함수이다.

(표 7) 보상채널 분산응용 프로그램의 함수
(Table 7) DAPP function list of SDRCN

함수명칭	기능
TCompCreate	순차 블록생성 성능측정함수
TCompRead	순차 블록읽기 성능측정함수
TCompUpdate	순차 블록수정 성능측정함수
TCompDelete	순차 블록삭제 성능측정함수
CcmCompCreate	병렬처리 블록생성 성능측정함수
CcmCompRead	병렬처리 블록읽기 성능측정함수
CcmCompUpdate	병렬처리 블록수정 성능측정함수
CcmCompDelete	병렬처리 블록삭제 성능측정함수
CcmCompCreateAsset	병렬처리 블록생성 함수
CcmCompReadAsset	병렬처리 블록읽기 함수
CcmCompUpdateAsset	병렬처리 블록수정 함수
CcmCompDeleteAsset	병렬처리 블록삭제 함수
CcmCompHistoryAsset	킷값에 해당하는 모든 블록정보 읽기 함수
CcmCompReadByDriverIDAsset	운전자별 블록정보 읽기 함수
CcmBanlanceQueryAsset	운전자별 보상잔액 검색
CcmBalanceHistoryAsset	운전자별 잔액 변동내역 검색

3.5 사고안전시스템(SDRS)

3.5.1 블록체인 네트워크 구성

사고안전시스템 블록체인 네트워크 시스템의 동작을 확인하기 위하여 표 8과 같이 2개의 조직과 조직별 2개의 피어노드, 블록을 생성하고 합의과정을 수행하는 오더러노드(Orderer Node)로 구성하였다.

(표 8) 노드 수준에서 조직의 구성
(Table 8) Organization at The Node level

노드	조직명	도메인
Orderer	Orderer	orderer.example.com
peer0	org1	peer0.org1.example.com
peer1	org1	peer1.org1.example.com
peer0	org2	peer0.org2.example.com
peer1	org2	peer1.org2.example.com

3.5.2 하드웨어 구성

하이퍼레저 패브릭 블록체인 시스템은 도커컴포즈(docker compose)를 이용하여 가상화 기반(Virtual Machine

Based)시스템으로 여러 대의 가상서버를 구성하였다. 본 연구에서 사용한 네트워크의 경우 피어노드마다 서버가 운영되어야 하며 CouchDB를 사용하는 경우 피어노드마다 데이터베이스 가상서버가 구동된다. 따라서 고성능서버의 사용이 블록체인의 성능을 높이는데 직접적인 관계가 있다. 본 연구에서 사용된 시스템의 사양을 표 9에 나타내었다.

(표 9) 시스템 사양
(Table 9) Hardware System Specification

구 분	사 양
CPU	Intel Core i9-990xe @3.0GHZ x36 core
GPU	Nvidia 1080 8GB memory
Memory	64G DDR4 2133MHZ
Storage SSD	1TB SAMSUNG 970 PRO
Hard disk	3TB TOSHIBA 7200RPM SATA 3.0 6Gb/S
Network	Ethernet 802.3bz

3.5.3 소프트웨어 구성

본 연구에서 구축한 블록체인 시스템은 높은 데이터 처리 능력을 요구하므로 고성능의 블록체인 네트워크 구성을 목적으로 개발언어를 선정하였다. 일반적으로 하이퍼레저 패브릭에서 체인코드개발에 사용되는 언어는 Go, Java, Node.js가 사용되며, 분산응용프로그램 개발은 Go, Python, Java 버전의 SDK가 제공된다. 개발언어별 특징으로 인하여 블록체인 처리속도에 차이가 있으며, Go언어가 가장 성능이 우수한 것으로 알려졌다. IBM Blockchain Platform의 Hyperledger Blockchain Performance Reports[13]에 따르면 Node.js 165TPS, Go언어 860TPS를 나타내고 있다. 본 연구에서는 Go언어를 체인코드 개발언어로 선정하였으며, 분산응용 프로그램 개발에도 Go언어를 사용하여 개발하였다. 하이퍼레저 패브릭 블록체인 개발을 위해 설치해야 하는 프로그램과 버전정보를 표 10에 나타내었다.

(표 10) 소프트웨어 구성
(Table 10) System Software Version

구 분	사 양
Operating System	Ubuntu 18.04 LST
fabric-contract-api-go	Version v1.1.1
Smart Contract Language	Go 1.6
Hyperledger Fabric Client golang SDK	V1.0
DAPP Language	Go 1.6
Remote Desktop S/W	Anydesk 6.3

4. 성능평가 및 분석

4.1 성능평가 개요

블록체인 시스템의 성능측정은 블록처리능력인 TPS 단위를 사용하여 측정한다. 분산응용 프로그램(DAPP)을 이용하여 네트워크에 접속하여 체인코드를 구동한다. 본 연구에서는 시스템의 성능측정을 위해 표 11과 같은 조건에서 프로그램을 수행하였다.

(표 11) 시스템 평가과정
(Table 11) System Evaluation Process

구분	내 용
데이터	5000 데이터셋(표4, 표6)
대상 네트워크	-SDCN 안전채널 -SDRCN 보상채널
비교대상	1)순차 블록생성 방법으로 성능측정 2)병렬처리 블록생성 방법으로 성능측정
실행방법	500개의 데이터 셋을 처리 후 평균시간을 측정하고 이를 10회 반복하여 평균시간 측정

4.2 사고안전 보상 시스템(SDRS)의 순차처리 성능평가

사고안전 보상 시스템(SDRS)에 접속하기 위한 분산응용프로그램을 Go언어를 이용하여 개발하였다. 실행 명령은 dapp이며, 외부 변수를 입력받아 실행하도록 하였다. 그림 7에는 순차 블록생성 함수 TDtgCreate를 실행하는 화면을 나타내었다.

```
qg@qg-System-Blockchain:~/fabric-samples/dapp$ ./dapp TDtgCreate 0 500 10
TDtgCreate 0 500 10
[fabsdk/core] 2022/01/11 19:11:59 UTC - cryptosuite.GetDefault -> INFO No default
cryptosuite found, using default SW implementation
0 execution time: 40.944594717s
1 execution time: 39.941797309s
2 execution time: 39.732684926s
3 execution time: 39.828781887s
4 execution time: 39.84580728s
5 execution time: 39.846572497s
6 execution time: 39.73116022s
7 execution time: 39.879785837s
8 execution time: 39.845254982s
9 execution time: 39.969008011s
total Time : 399565447666ns  Iterations : 10  average Time : 39.9565447666 sec
Tps: 12.513595
qg@qg-System-Blockchain:~/fabric-samples/dapp$
```

(그림 7) 블록생성 명령 실행화면
(Figure 7) Screen Capture of the Block Creation Command

첫 번째 매개변수 TDtgCreate 안전채널에 블록 데이터를 생성하는 체인코드를 호출하는 함수로 3개의 매개변

수를 사용한다. 두 번째 매개변수 0은 블록의 시작번호를 나타내며, 세 번째 500은 500개의 블록처리를 순차적으로 실행하는 값이며, 네 번째 매개변수 10은 500개의 블록처리를 10번 반복하여 실행하도록 지정하는 값이다. 마지막 라인에 출력되는 값은 프로그램 실행시간과 500개의 블록처리에 소요되는 평균시간을 나타내며, 5,000개의 블록처리에 초당 12.51개의 블록을 처리한 결과를 12.51TPS로 나타났다.

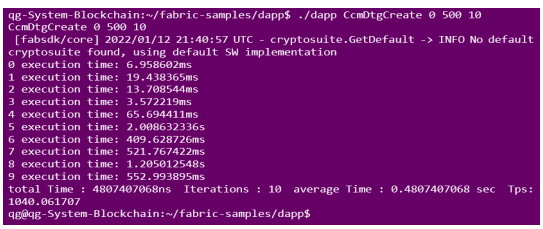
안전채널과 보상채널의 블록처리 성능을 평가하기 위해서는 블록의 생성, 읽기, 변경, 삭제의 4가지 기본기능을 평가하여야 하며, 그 결과는 표 12에 나타내었다.

(표 12) 순차 블록처리 성능
(Table 12) Sequential Block Transaction Performance

명령어	총수행시간(초)	평균 값(초)	TPS
안전채널 블록생성	399.56	39.95	12.51
안전채널 블록읽기	54.34	5.43	91.99
안전채널 블록수정	370.23	37.02	13.50
안전채널 블록삭제	367.98	36.79	13.58
보상채널 블록생성	399.61	39.96	12.51
보상채널 블록읽기	54.42	5.44	91.87
보상채널 블록수정	368.70	36.87	13.56
보상채널 블록삭제	542.54	54.25	9.21

4.3 사고안전 보상 시스템(SDRS)의 병렬처리 성능평가

블록처리 성능향상을 위한 병렬처리기법은 36개의 코어 프로세스를 모두 사용하도록 하였으며, 높은 사용률을 유지하도록 Go언어의 동시처리 지원 도구인 CCM (Concurrent Control Manager)을 이용하여 구현 하였다. 병렬처리 속도측정을 위한 블록생성 명령 CcmDtgCreate의 실행화면을 그림 8에 나타내었다.



(그림 8) 블록생성 명령 실행화면
(Figure 8) Screen Capture of the Block Creation Command

CcmDtgCreate 함수의 실행방법은 순차성능평가에서와 같이 실행되었으며 5,000개의 블록처리에 초당 1,040개의 블록을 생성하여 1,040TPS의 처리속도를 나타내었다. 표 13은 채널별 성능측정 값이다.

(표 13) 병렬 블록처리 성능
(Table 13) Concurrent Block Transaction Performance

명령어	총수행시간(초)	평균값(초)	TPS
안전채널 블록생성	4.80	0.48	1040.06
안전채널 블록읽기	2.00	0.20	2496.10
안전채널 블록수정	2.41	0.24	2067.15
안전채널 블록삭제	2.26	0.22	2206.22
보상채널 블록생성	5.23	0.52	955.27
보상채널 블록읽기	1.98	0.19	2515.27
보상채널 블록수정	2.79	0.27	1792.08
보상채널 블록삭제	2.68	0.26	1863.97

4.4 사고안전 보상 시스템(SDRS)의 순차처리와 병렬처리 성능평가 결과 비교

본 연구에서는 블록체인 네트워크에서 순차처리와 병렬블록처리 결과를 비교하기 위하여 블록의 읽기, 쓰기, 변경, 삭제 기능에 대한 성능을 비교 평가했다. 그 결과, 표 14와 같이 블록 읽기는 12.51TPS에서 1,040TPS로 83배 이상 향상되었으며, 블록 삭제의 경우 9.21TPS에서 1,863.97TPS로 200배 이상의 성능향상을 나타내고 있어 병렬처리를 이용한 가속 적용 방법의 효용성을 입증하였다.

(표 14) 순차 대 병렬처리 성능 비교
(Table 14) Sequential vs Concurrent Transaction Performance

명령어	가속 미적용	가속 적용	속도 비교(배)
안전채널 블록생성	12.51	1040.06	83.14
안전채널 블록읽기	91.99	2496.10	27.13
안전채널 블록수정	13.50	2067.15	153.12
안전채널 블록삭제	13.58	2206.22	162.46
보상채널 블록생성	12.51	955.27	76.36
보상채널 블록읽기	91.87	2515.27	27.38
보상채널 블록수정	13.56	1792.08	132.16
보상채널 블록삭제	9.21	1863.97	202.39

5. 결 론

블록체인 시스템의 많은 장점에도 불구하고 낮은 블록 처리속도로 인하여 실용화에 제약이 따른다. 본 연구에서도 화물차를 대상으로 한 블록체인 시스템으로 수만대의 차량을 처리하는데 10TPS 내외의 속도로는 실제 활용이 불가능하다. 따라서 본 연구에서는 속도향상을 위해서 Go언어의 병렬처리기법을 이용하여 2,500개의 프로세스를 유지하는 가속 루틴을 개발하여 블록체인의 성능을 향상시켰다. 그 결과 블록읽기 성능은 83배 이상 향상되었으며, 블록 삭제의 경우 200배 이상의 성능향상을 나타내고 있다. 이러한 성능향상이 가능했던 것은 하이퍼레저 패브릭 버전이 2.2로 업그레이드되면서 동시처리에 강력한 성능을 발휘하는 Go언어 SDK[14]가 제공되었기 때문이며, 동시처리 프로세스를 모니터링하여 프로세스가 끝나면 즉시 새로운 동시처리 프로세스를 수행시켜 높은 병렬처리 수준을 유지하도록 한 알고리즘 적용으로 가능하였다. 본 연구의 한계는 하나의 서버에 가상머신을 설치하여 다수의 가상서버가 수행되도록 시스템을 구성하므로 시스템의 성능이 하드웨어 성능한계에 수렴되는 특성이 있다. 이러한 성능한계를 극복하기 위해서는 클라우드 기반의 알고리즘 개발이 필요하며, 본 연구결과를 토대로 화물차 안전보상을 위한 실증연구를 지속할 예정이다.

참고문헌(Reference)

[1] 사업용 차량 디지털운행기록계 장착 의무화, 국토해양부 보도자료, 2010.

[2] 이유미, “태호·유찬이법 국회통과”, 연합뉴스, http://yna.co.kr/view/AKR20200429213000001, 2020.

[3] Nakamoto, Satoshi "Bitcoin P2P e-cash paper", Archived from the original on 28 December 2012. https://archive.is/DUcfX (31 October 2008)

[4] Adrian Gallagher, Litecoin Project, https://blog.litecoin.org , 2022.

[5] Ethereum Foundation, https://ethereum.org/en, 2022.

[6] Ripple Company, https://ripple.com, 2022.

[7] https://www.bitcoincash.org, 2022.

[8] https://eos.io, 2022.

[9] J.S. Park, J.Y. Park, S.M. Choi, J.T. Oh, K.Y. Kim, “Past, Present and Future of Blockchain Technology“, Electronics and Telecommunications Trends, Vol 33, No.6, pp.139-153, 2018. https://doi.org/10.22648/ETRI.2018.J.330614

[10] Su-Hwan Bae, Sun-Ok Cho, Yong-Tae Shin, “A System Recovery Using Hyper-ledger Fabric BlockChain”, Vol12, No.2, 2019. https://doi.org/10.17661/jkiiect.2019.12.2.155

[11] Shafkat Islam, Shahriar Badsha, Shamik Sengupta, “A Light-weight Blockchain Architecture for V2V”, IEEE International Smart Cities Conference (ISC2), 2020. http://doi.org/10.1109/ISC251055.2020.9239055

[12] Yongbae Kim, Juyong Back, Jongweon Kim, “A Tamper-Resistant Algorithm Using Blockchain for the Digital Tachograph”, Electronics, 2021. https://doi.org/10.3390/electronics10050581

[13] Caliper Benchmarks, “Hyperledger Blockchain Performance Reports”, https://hyperledger.github.io/caliper-benchmarks, 2022.

[14] Hyperledger Fabric Client SDK for Go, 2022. https://pkg.go.dev/github.com/hyperledger/fabric-sdk-go

◐ 저 자 소 개 ◑

김 용 배(Yong-bae Kim)
1987년 충남대학교 전기공학교육 학과(공학사)
1991년 충남대학교 대학원 전기공학과(공학석사)
1995년 충남대학교 대학원 전기공학과(공학박사 수료)
2016년~현재 상명대학교 산학협력단 연구원
관심분야 : 신호처리, 블록체인, 데이터베이스, 인공지능
E-mail : ybkimdb@gmail.com

백 주 용(Juyong Back)
2019년 한양대학교 융합산업대학원 디자인매니지먼트과(석사수료)
1996~2018년 스퍼트커뮤니케이션즈 대표이사
2016~현재 주식회사 퀀텀게이트 대표이사
관심분야 : I2V, C-ITS, 데이터베이스, 블록체인
E-mail : v@quantumgate.co.kr

김 종 원(Jongweon Kim)
1989년 서울시립대학교 전자공학과(공학사)
1991년 서울시립대학교 대학원 전자공학과(공학석사)
1995년 서울시립대학교 대학원 전자공학과(공학박사)
2009년~현재 상명대학교 소프트웨어융합학부 지능IOT융합전공 교수
관심분야 : 디지털신호처리, 멀티미디어 보안, 컴퓨터 비전, 인공지능
E-mail : jwkim@smu.ac.kr