

전자상거래 로열티 서비스를 위한 시스템 제공자 설계 및 구현

(Design and Implementation of the System Provide of the Loyalty Service for Electronic Commerce)

홍재영[†] 김기영^{**} 이영석[†] 최훈^{***}
(Jae-Young Hong) (Ki-Young Kim) (Young-Seok Lee) (Hoon Choi)

김영국^{****} 홍승재^{****}
(Young-Kuk Kim) (Seungjae Hong)

요약 로열티는 상품 구입의 대가로 고객에게 지급하는 보너스이고, 로열티 시스템은 스마트 카드를 이용하여 이러한 로열티의 이동을 제어하는 시스템이다. 기존 가맹점 위주의 로열티 시스템은 기업, 상점 별로 독자적으로 운영되어 시스템간 호환성이 없고 한 상점에서 받은 로열티를 다른 상점에서 사용하는데 제약이 있다. 따라서 본 논문에서는 기존 가맹점 위주의 로열티 시스템의 범위를 확대해서, IC 카드를 기반으로 여러 기업들에 공동으로 적용되고 로열티 판매, 결제, 관리를 온라인으로 처리하는 시스템을 개발하였다. 이 논문은 로열티 시스템의 핵심 요소인 NAC, TCS, TMS의 기능 구현 사항과 로열티 시스템의 통신 프로토콜에 대해 기술한다.

Abstract Loyalty is a bonus point that a customer earns in reward of purchasing goods of a shop or a manufacturer. The management of loyalty such as issuing, accounting, reimbursing is carried out by the loyalty system. Traditional loyalty systems were proprietary. As a result, the systems are not interoperable with each other. However, our loyalty system allows a group of merchants join together and use the same IC card to operate their own loyalty programs. Data and counters can be shared or totally isolated, using secret codes and keys. The loyalty pool system communicates with the loyalty service providers over the Internet and performs the management of loyalty in fully automated manner. This paper describes the functional architecture and internal behavior of the loyalty pool system along with the communication protocols.

1. 서론

· 본 논문은 과학재단지정 충남대학교 소프트웨어연구센터의 지원으로 수행된 과제(과제번호 : 99-11-12-99-A-1)의 결과임

[†] 비회원 : 충남대학교 컴퓨터공학과
jyhong@cc.cnu.ac.kr
yslee@cc.cnu.ac.kr

^{**} 비회원 : 한진정보통신 기술연구소 연구원
gykim99@hist.co.kr

^{***} 종신회원 : 충남대학교 컴퓨터공학과 교수
hchoi@comcng.chungnam.ac.kr

^{****} 정회원 : 충남대학교 정보통신공학부 교수
ykim@cs.chungnam.ac.kr

^{*****} 비회원 : 씨고마테크(주) 연구소 소장
sungjae@netsgo.com

논문접수 : 1999년 11월 9일
심사완료 : 2000년 2월 29일

전자상거래는 기업간 또는 기업과 개인, 정부간에 컴퓨터 네트워크를 통해서 이루어지는 다양한 거래를 의미하는 것으로 정보기술의 통합적 활용을 통하여 기업의 경영 효율을 높이고 국제시장에서 경쟁력을 강화할 수 있는 수단이다[1]. 이것이 활성화 되기 위해서는 여러 가지 선결해야만 하는 일들이 존재한다. 그 중에서도 가장 핵심적인 요소로 떠오르고 있는 것이 전자지불시스템(Electronic Payment System)이다. 전자지불이란 기존의 화폐 개념을 네트워크 상으로 옮겨 디지털화 한 무형의 화폐 또는 지불수단을 의미한다[2]. 이러한 전자지불 수단의 하나인 로열티(loyalty)는 상거래에 부수되어 고객에 대한 충성도를 높이는 방식으로서, 이미 쿠폰

(coupon), 할인카드(discount card), 포인트카드(point card), 복권(bonus card) 등의 형태로 많이 사용되고 있다[3]. 이러한 로열티는 전자상거래에서도 더욱 일반화 될 것이며 현재와는 다른 새로운 기술이 적용된 형태일 것이다.

로열티가 편리하고 안전하게 전자상거래에서 운영되기 위해서는 전반적인 네트워크 보안과 서버 보안 문제를 해결해야 하고, 고객이 로열티를 보관하고 거래하는 수단으로 사용하기 위해서 IC카드(스마트 카드)[4][5][6][7]와 같은 휴대 가능한 보안도구가 필수 불가결하다. 또한 기존의 로열티 시스템은 기업, 상점별로 독자적으로 운영되어 시스템간 호환성이 없고 한 상점에서 받은 로열티를 다른 상점에서 사용하는데 제약이 있다. 전자상거래에서의 로열티 시스템은 여러 기업들에 공동으로 적용되고 로열티 판매, 결제, 관리를 온라인으로 처리할 수 있는 시스템이어야 한다.

국내의 로열티 서비스는 개별 업체 중심으로 이루어지고 있어서 고객측면에서 볼 때 다양한 서비스에 이용할 수 없고, 종이 등의 쿠폰은 관리상 어려움이 따르기 때문에 실제 효과를 보지 못하고 있다. 이에 비해, 외국의 로열티 서비스는 점차 활성화되고 있다. 프랑스에서는 Datagram이라는 스마트 카드를 이용한 로열티 전문 회사가 등장하여 해외로 진출하고 있다[8]. 홍콩에서는 Compass 시스템이란 이름으로 은행들이 연합하여 IC카드 기반 로열티 시스템을 제공하고 있으며, 싱가포르에서도 1999년 5월부터 IC 카드 기반한 NEW(Never Ending Wishes) 시스템이 제공되고 있다[9]. 미국에서는 MyPoints 시스템이 인터넷을 통한 전자상거래 고객에게 로열티를 제공하고 있으며[10], 그 밖에 독일 G&D의 STARCOIN system[11], 캐나다의 AIR MILES[12] 등에서도 로열티 시스템을 개발하거나 이미 서비스를 제공하고 있다.

본 논문에서는 전자상거래 활성화를 위한 노력의 일환으로 기존의 가맹점 위주로 제한된 로열티 서비스를 개선하여, IC 카드에 기반을 두고 여러 가맹점들이 통합된 고객 위주의 Loyalty System을 개발하였다. 이 논문에서는 Loyalty System 중 시스템 제공자에 대해 소개한다. 다중 연결을 관리하며 가맹점과의 통신을 담당하는 NAC, 트랜잭션의 백업을 관리하는 TLS, 데이터를 분리하여 해당 시스템 엔터티에게 전송하는 TCS, 그리고 로열티 처리를 담당하는 TMS에 대해 설명하고, 이 시스템에서의 통신 프로토콜을 기술한다.

논문의 구성은 다음과 같다. 2절에서는 로열티 시스템의 개요를 간단히 소개하고, 3절에서는 시스템 제공자인

Loyalty Pool 시스템의 기능 구조를 제시하며, 4절에서는 로열티 시스템에서 사용되는 프로토콜의 TLV(Tag Length Value) 메시지를 정의하고, 5절에서 구현 및 시험에 대해서 살펴본 후 결론을 맺도록 한다.

2. 로열티 시스템 개요

로열티 시스템은 시스템 제공자, 서비스 제공자, 고객 IC 카드, 카드 발급 시스템 및 관리자 시스템으로 구성된다.

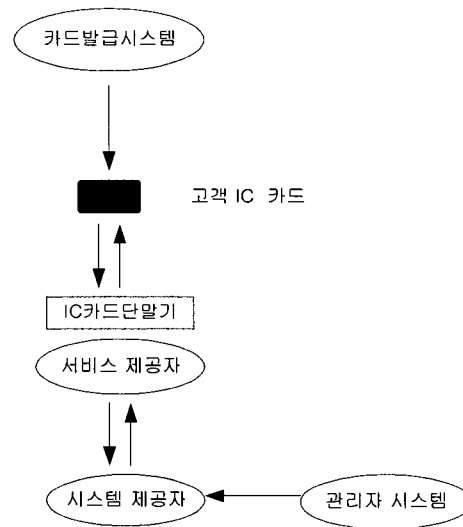


그림 1 시스템 구성

본 연구에서 구현한 로열티 시스템은 스마트 카드에 기반하고 있기 때문에, 우선 IC 카드를 발급할 수 있는 카드 발급 시스템이 필요하다. 만들어진 카드는 IC 카드 단말기를 통해 서비스 제공자와 인터페이스하게 된다. 서비스 제공자는 회사, 상점과 같은 로열티 시스템 가맹점으로서 고객을 상대로 로열티를 지급한다.

로열티 시스템의 핵심은 시스템 제공자인 Loyalty Pool이다. 이것은 서비스 제공자와 인터페이스하여 로열티를 공유할 수 있는 수단을 제공하는데 Loyalty Pool에서는 보안 처리, 사용자의 질의에 대한 트랜잭션 관리, 트랜잭션 백업, 외부 금융 서비스와의 인터페이스, 가맹점 정보 DB(Database)와의 인터페이스를 담당한다. 또한 시스템 제공자는 관리자 시스템에 의해 관리된다.

Loyalty Pool은 그림 2와 같이 NAC, SMS, TCS, TLS, RAS의 각 시스템 엔터티로 구성되어 있다. 각 시스템 엔터티의 역할을 살펴보면 다음과 같다.

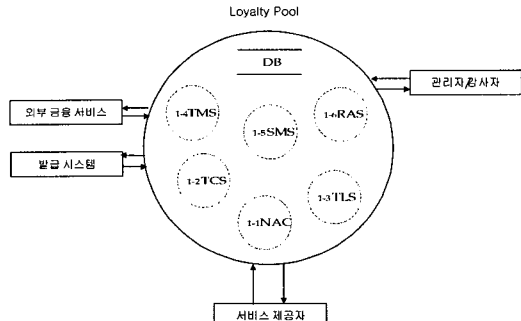


그림 2 Loyalty Pool 기능 엔터티

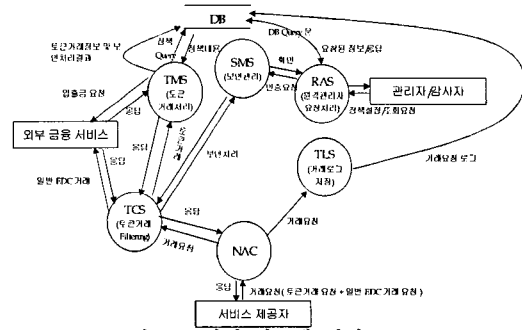


그림 3 로열티 시스템 서비스 DFD

- NAC(Network Access Controller): 서비스 제공자와 시스템 제공자 사이의 다중 연결(multi connection)관리, 통신처리, CRC(Cyclic Redundancy Check) 체크
- SMS(Security Management System) : 서비스 제공자, 시스템 제공자 사이의 메시지 암호화, 복호화에 관련된 기능 제공
- TCS(Token Capture System) : 로열티 관련 데이터와 EDC(Electronic Data Capture) 데이터를 분리하여 로열티 관련 데이터는 TMS로, EDC 관련 데이터는 VAN(Value Added Network)으로 중계
- TMS(Token Management System) : 로열티 토큰 구매, 판매 등과 같은 로열티 관련 서비스 제공, DB와의 인터페이스
- TLS(Transaction Log Server) : 안정성을 위해 시스템 제공자로 보내진 모든 트랜잭션을 백업
- RAS(Remote Access Server) : 관리자나 감사자의 접근 수단을 제공

본 논문에서 TMS와 TCS는 공통되는 기능이 존재하므로 같은 서버에 구현하였고, NAC와 SMS, TLS, RAS는 별도의 서버에 구현하였다.

3. 시스템 제공자

3.1 시스템 서비스 DFD

서비스 제공자 측면에서 로열티 시스템이 제공하는 서비스는 로열티 토큰의 구매, 판매, 환불, EDC 거래, 정보 조회가 있다. 이 서비스들을 처리하기 위한 기능 엔터티들 간의 DFD(Data Flow Diagram)는 다음과 같다.

서비스 제공자로부터 토큰 구매, 판매, 환불, 정보 조회 중 하나의 서비스 요청이 NAC로 보내지면, NAC는 요청 메시지를 받아 CRC를 검사한 뒤에 올바른 메시지라고 판명되면 TLS와 TCS로 보낸다. CRC 검사를 통해 메시지 형태의 이상 유무를 확인할 뿐만 아니라 이 메시지가 등록된 서비스 제공자로부터 보내온 것인지 여부를 조사한다. TCS에서는 수신된 메시지를 SMS로 보내 복호화하고 로열티 관련 데이터와 EDC 관련 데이터를 분류한 후에, 로열티 데이터는 TMS로 전송하고 EDC 관련 데이터는 VAN을 통해 외부 금융 서비스로 전달한다. 따라서 본 로열티 시스템은 서비스 제공자들이 별도의 통신 수단을 구비하지 않아도 외부 VAN 사업자와 통신할 수 있는 서비스도 제공한다.

TMS는 메시지를 분석하여, 서비스 제공자, 즉 가맹점별 로열티 서비스 정책이 저장된 정보 DB를 조회한 후 이에 따라 로열티 처리를 수행한다. SMS는 RAS와 TMS, TCS의 요청에 따라 메시지의 암호화, 사용자 인증 등을 수행한다.

3.2 기능 엔터티 간 IFD

3.2.1 토큰 구매 절차

3.1절에서 제시한 서비스와 그에 따른 DFD를 기반으로 로열티 토큰 구매시의 Loyalty Pool 엔터티들 간의 IFD(Information Flow Diagram)와 시스템 엔터티 내부 동작을 살펴보면 다음과 같다.

로열티 서비스 제공자인 가맹점이 시스템 제공자인 Loyalty Pool로 토큰 구매를 요청하려면, 터미널을 통해 토큰 구매 요청 메시지를 만들고 암호화시킨 뒤 4byte의 CRC 코드를 덧붙여서 Loyalty Pool로 전송한다. Loyalty Pool의 NAC는 암호화된 메시지를 받아서 CRC만을 체크해 본 뒤 TLS와 TCS에게 전송한다. TLS는 현재 트랜잭션을 백업하고, TCS는 암호화된 메시지를 SMS로 보내 복호화된 메시지를 얻어온 다음,

메시지를 분석해서 서비스 요청을 처리하기 위한 정보를 추출한 뒤 구조체에 저장해 놓고, TMS를 호출한다. TMS는 TCS에서 추출한 정보를 기반으로 DB와의 인터페이스를 통해 트랜잭션을 처리하고, 응답 메시지를 만들어서 TCS에게 전송한다. TCS는 다시 SMS와의 인터페이스를 통해 메시지를 암호화 시킨 후에 NAC를 통해 서비스 제공자로 메시지를 전송하게 된다. 이러한 과정을 통해 토큰 구매 트랜잭션이 완료되도록 구현하였다.

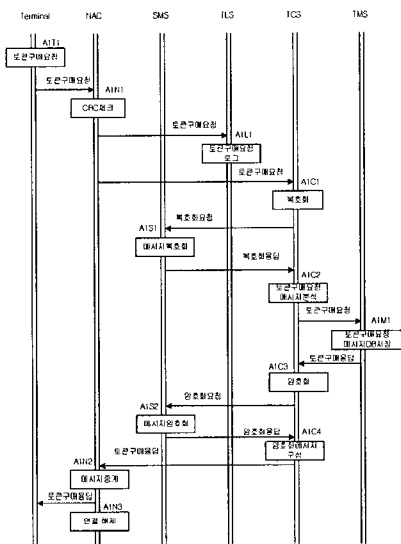


그림 4 토큰 구매의 IFD

토큰 판매와 환불, 정보 조회의 IFD는 토큰 구매 IFD에 나타난 토큰 구매가 토큰 판매와 환불 그리고 정보 조회로 바뀌어진다는 점을 제외하면 토큰 구매의 IFD와 동일하다.

3.2.2 EDC 거래절차

3.1절에서 제시한 서비스와 그에 따른 DFD를 기반으로 EDC 거래시의 Loyalty Pool 엔터티들 간의 IFD(Information Flow Diagram)를 그려보면 다음과 같다.

EDC 거래는 TMS가 아닌 외부 금융 서비스와 인터페이스 한다는 점이 다른 서비스들과 다르다. NAC로부터 전송되어진 데이터가 EDC 관련 데이터 인지 혹은 로열티 관련 데이터인지 TCS에서 판별하여, EDC 데이터 일 경우에는 VAN을 통해 외부 금융 서비스로 전송하게 된다.

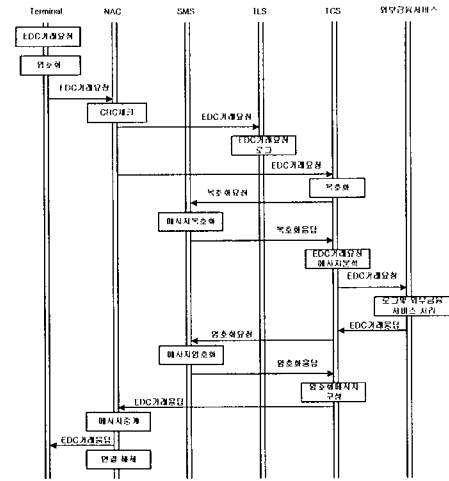


그림 5 EDC 거래시의 IFD

3.3 기능 엔터티의 내부 동작 절차

토큰 구매시의 내부 동작 절차(그림 4 참조)를 살펴보면 다음과 같다.

- System Entity Internal Action

A1T1(토큰구매요청)

1. 터미널을 통해 토큰 구매 요청 명령
2. 요청을 정의된 TLV 포맷으로 변환
3. TLV화된 메시지 중 Value만 추출하여 암호화
4. 암호화된 메시지에 첫 번째 필드를 '9F00'으로 두 번째 필드를 암호화된 Value의 길이로 추가
5. CRC 필드를 TLV 메시지에 추가
6. 연결 설정 이후 토큰 구매 요청 메시지를 NAC로 전송

A1N1(CRC 체크)

1. 터미널로부터 오는 메시지를 수신
2. 받은 메시지의 CRC 비트를 검사한 뒤 메시지에서 CRC 비트 제거
3. TLS와 TCS에게 메시지 전송

A1N2(메시지 중계)

1. TCS로부터 메시지 수신
2. 터미널에게 메시지 전송

A1N3(연결 해제)

1. 연결 설정을 해제

A1S1(메시지 복호화)

1. TCS로부터 암호화된 메시지 수신
2. 메시지를 복호화 후 TCS에게 전송

A1S2(메시지 암호화)

1. TCS로부터 암호화되지 않은 메시지 수신
2. 메시지를 암호화 후 TCS에게 전송

A1L1(토큰구매요청로그)

1. NAC로부터 메시지 수신
2. 트랜잭션을 저장

A1C1(복호화)

1. NAC로부터 메시지 수신
2. 수신한 TLV 메시지 중 첫 번째 필드인 '9F00'과 두 번째 필드인 전체 길이를 제거한 뒤 Value만 복호화를 위해 SMS에게 전송

A1C2(토큰 구매 요청 메시지 분석)

1. SMS로부터 복호화된 메시지를 수신
2. 수신한 메시지에 첫 번째 필드로 '9F00'을 두 번째 필드로 복호화된 Value의 길이를 계산하여 추가
3. 메시지를 분석하여 공유 메모리 구조체에 저장
4. TMS에게 토큰 구매 요청

A1C3(암호화)

1. TMS로부터 메시지 수신
2. 수신한 메시지 중 Value만을 추출하여 SMS로 전송

A1C4(암호화 메시지 구성)

1. SMS로부터 암호화된 메시지를 수신
2. 수신한 메시지에 첫 번째 필드로 '9F00'을 두 번째 필드로 암호화된 Value의 길이를 계산하여 추가
3. NAC로 메시지 전송

A1M1(토큰 구매 요청 메시지 DB 저장)

1. TCS로부터 토큰 구매 요청 수신
2. 공유 메모리 구조체로부터 메시지 정보를 얻음
3. 정보를 바탕으로 DB 저장, 갱신하여 토큰 구매, 환불 처리
4. TLV 포맷으로 응답 메시지 구성 후 TCS 메시지 전송

4. 프로토콜 메시지

TLV는 Tag, Length, Value로 이루어진 정보 전송 형식으로서 ASN.1의 Basic Encoding Rules에 정의된 것이다[13]. 본 연구에서는 로열티 시스템의 시스템 제공자와 서비스 제공자간 통신을 위해 TLV에 기반한 프로토콜을 정의하였다. Tag는 4바이트의 고정길이 필드로서 Value의 타입을 구분시켜 주고, Length는 4바이트 고정길이 필드로 Value 필드 길이 정보를 가지고 있다. 그리고 Value는 상대 시스템에게 전달하기 위한 실제적인 값으로 크기에 제한을 두지 않았다.

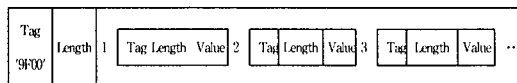


그림 6 토큰 구매시의 TLV 메시지

하나의 시스템에서 위와 같이 TLV 메시지를 구성하여 다른 시스템에게 전송하면, TLV 프로토콜은 먼저 Tag를 확인한 뒤, Length만큼 오프셋(offset)을 조정하여 Value 값을 추출한다. 이렇게 TLV를 정해 놓음으로써, 수신자는 하나의 스트링을 만들어서 보내고, 송신자는 스트링을 받아 다시 분석할 수 있게 된다.

그림 6에서 보는 것처럼, 본 논문에서 구현한 시스템의 TLV는 Value에 다수의 또 다른 TLV가 포함될 수 있는 구조화 형식(constructed type)을 사용하였다.

4.1 토큰 구매 시의 TLV 메시지

토큰 구매 시 서비스 제공자로부터 NAC로 전송되는 메시지들의 Tag값들은 다음 표와 같다.

표 1 토큰 구매시 이용되는 Tag 종류

Tag	의 미
9F00	전송 메시지의 dbud을 결정
9F16	상점 식별번호
9F18	상점 전화번호
9F10	로열티 풀(Pool) 식별번호
9F42	표준 시간
9F23	트랜잭션 날짜
9F21	트랜잭션 시간
9F50	메시지 타입
9F60	Token Certificate
9F80	은행 식별코드
9F81	은행 계좌번호

표 2 Token Certificate의 구성

Value	길이	의미
Version	3byte	토큰 버전
Serial Number	12byte	토큰 구별하기위한 값
Signature	5byte	전자 서명정보
Issuer	8byte	제공자
Validity	16byte	유효 날짜
Token Type	3byte	토큰 타입
Token Amount	5byte	토큰 양
TTC	3byte	Token Transaction Count
CTC	3byte	Card Transaction Count

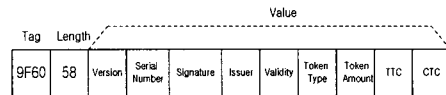


그림 7 실제 Token Certificate 모습

[표 1]에서 Tag가 9F60인 Token Certificate는 [표 2]와 같이 58byte로 구성된다.

토큰 구매 시 Token Certificate 메시지와 같이 하나의 Tag를 정해 놓고, 고정길이 Value의 여러 정보를 하나로 묶어서 메시지를 구성한다면, Tag의 종류를 줄이고 네트워크 트래픽 양을 줄이는 장점이 있다. 그러나 추가해야 할 Tag가 생기거나, 메시지에 정보를 추가/삭제해야 할 때 전체 시스템을 변경해야 한다는 단점이 있다. 따라서 시스템의 유연성(flexibility)과 확장성

(scalability)을 고려해서 변경되지 않을 Value들을 모아 하나의 Tag으로 구성시키고, 그 이외의 것들은 Value 당 별도의 Tag을 할당하여 정의하였다.

5. 구현 및 시험

서비스 제공자 측면, 즉 터미널에서 제공하는 토큰 구매, 토큰 판매, 토큰 환불, EDC 거래, 자료 조회는 Visual C++6.0을 사용하여 Windows 95/98에서 구현하였으며, 시스템 제공자 측면, 즉 NAC, SMS, TCS, TLS, TMS는 C++를 사용하여 Redhat Linux[14]에서 구현하였다. 또한 통신은 TCP/IP에 기반한 Socket[15][16]을 이용하여 구현하였다.

NAC는 서비스 제공자와 시스템 제공자 사이의 다리 역할을 담당한다. NAC는 여러 가맹점으로부터 서비스 요청을 받아서 해당 시스템 엔터티에게 중계하는 역할을 담당하므로 복수의 연결을 동시에 유지하는 것이 필요하다. 본 연구에서는 이것을 위해 다음과 같이 구현하였다. 우선 여러 포트를 초기화시키고, 서비스 제공자로부터 연결 요청이 오기를 기다린다. 연결 요청이 들어오면, 초기화시켜 놓은 포트 중 하나를 선택하여 포트를 할당해주고 연결을 설정한다. 다른 가맹점으로부터 연결 요청이 들어오면 초기화 되어진 포트 중에 사용되고 있지 않은 것을 할당받아 또 다른 연결을 설정하게 된다. 모든 서비스가 끝난 뒤에는 포트를 다시 초기화시켜 주

어 재사용할 수 있게 한다. 이런 방법으로 원하는 수만큼의 연결을 유지하도록 하였다.

서비스 제공자가 보낸 메시지가 NAC를 거쳐 TCS에게 전달되면, 이 후 TCS 뿐만 아니라 TMS, SMS 등 시스템 제공자의 다른 기능들도 메시지에 들은 정보를 이용하기 때문에 각 기능들 간의 신속한 메시지 정보 교환이 성능을 좌우하게 된다. 기능들 간의 통신 방법으로 메시지 교환 방식과 공유 메모리에 의한 방식을 이용할 수 있으나 공유 메모리 방식이 더 효율적이므로 본 구현에서도 이 방식을 채택하였다. NAC로부터 전송된 메시지를 TCS가 분석하여 공유 구조체에 저장시킨 뒤, TMS가 구조체를 이용하여 DB와 인터페이스를 할 수 있도록 구현하였다. TCS와 TMS가 사용하는 구조체를 정의하면 다음과 같다.

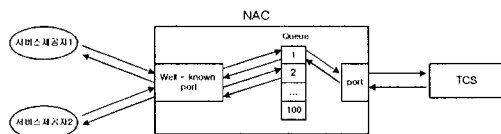


그림 8 다중 연결 유지

- 토큰 구매, 판매 환불시

```
typedef struct{
    unsigned int shop_id, pool_id;
    unsigned int transaction_id, transaction_year;
    unsigned int transaction_month, transaction_hour;
    unsigned int transaction_minute, transaction_second;
    unsigned int message_type;
    Token_Certificate token_info;
    unsigned int bank_id;
    unsigned char bank_account[20];
    Condition_Retrieve condition_data;
    Log_Retrieve log_data;
} real_data;
```

- 조건별 자료 조회

```
typedef struct{
    unsigned int condition_begin_year, condition_begin_month;
    unsigned int condition_begin_day, condition_end_year;
    unsigned int condition_end_month, condition_end_day;
    unsigned int condition_message_type;
} Condition_Retrieve;
```

- 날짜별 자료 조회

```
typedef struct{
    unsigned int log_begin_year, log_begin_month;
    unsigned int log_begin_day, log_end_year;
    unsigned int log_end_month, end_day;
} Log_Retrieve;
```

```
FD_ZERO(&readfds);
FD_SET(server-sockfd,&readfds);
while(1) {
    result = select(...);
    //초기화 시켜 놓은 포트 중 하나 선택
    for(fd=0; fd < FD_SETSIZE; fd++) {
        if(FD_ISSET(fd,&testfds))
            새로운 요청이므로 연결;
        else if
            모든 작업이 끝났으므로 포트 반환;
        else
            연결이 된 후, 해당 작업 처리;
    }
}
```

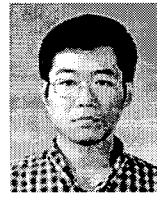
구현한 로열티 시스템은 그림 9에 보인 환경 상에서 시험을 수행하였다. 먼저 정상 상황과 비정상 상황들에

- [4] <http://www.visa.com/cgi-bin/vee/nu/main.html?2+0>
- [5] <http://www.java.sun.com>
- [6] <http://www.smartcardsys.com>
- [7] 김종률, "전자상거래에서 IC카드 소프트웨어 및 활용방안 연구", 정보화 저널, 제 3 권 제 4 호, 한국전산원, 1996
- [8] <http://www.datagrams.com>
- [9] <http://www.new.com.sq/>
- [10] <http://www.mypoints.com/>
- [11] <http://www.gdm.de/>
- [12] <http://www.airmiles.ca>
- [13] A.Tang, S.Scoggins, "Open Networking with OSI," Prentice-Hall, 1992
- [14] Neil Matthew, Richard Stones, "The beginning LINUX Programming," Wrox, 1999
- [15] W.Richard Stevens, "UNIX Network Programming," Prentice-Hall, 1991
- [16] 김화중, "컴퓨터 네트워크 프로그래밍", 홍릉과학출판사, 1008.3



김 영 국

1985년 서울대학교 계산통계학과 졸업. 1987년 서울대학교 계산통계학과 석사. 1995년 버지니아대학교 컴퓨터과학과 박사. 1995년 VTT (Technical Research Centre of Finland) 방문연구원. 1995년 SINTEF Telecom & Informatics, Norway 방문연구원. 1996년 ~ 현재 충남대학교 정보통신공학부 조교수. 관심분야는 실시간데이터베이스시스템, 전자상거래시스템, 분산정보시스템 등



홍 승 재

1994년 콜로라도텍니칼대학 전산과학 졸업. 1995년 LG정보통신(주) 소프트웨어연구소 연구원. 1996년 한국신용통신(주) 연구개발실 연구원. 1997년 ~ 1998년 BR네트웍 연구개발실 실장. 1999년 ~ 현재 씨그마크(주) 연구소 소장. 관심분야는 스마트카드, 전자상거래 보안 등



홍 재 영

1999년 충남대학교 공과대학 컴퓨터공학과 졸업. 1999년 ~ 현재 충남대학교 공과대학 컴퓨터공학과 석사과정. 관심분야는 전자상거래, 이동통신, 시스템 모델링 등



김 기 영

1998년 충남대학교 컴퓨터공학 교육학과 졸업. 2000년 충남대학교 컴퓨터공학과 석사. 2000년 ~ 현재 한진정보통신 기술연구소 연구원. 관심분야는 분산시스템, CORBA, XML 등



이 영 석

1992년 충남대학교 컴퓨터공학과 졸업. 1994년 충남대학교 컴퓨터공학과 석사. 1994년 ~ 1997년 LG정보통신(주)중앙연구소 연구원. 1997년 ~ 현재 충남대학교 컴퓨터공학과 박사과정. 관심분야는 분산시스템, 가상사설망, 이동컴퓨팅 등



최 훈

1983년 서울대학교 컴퓨터공학과 졸업. 1990년 듀크대학교 전산학 석사. 1993년 듀크대학교 전산학 박사. 1983년 ~ 1996년 한국전자통신연구원 책임연구원. 1996년 ~ 현재 충남대학교 컴퓨터공학과 조교수. 관심분야는 분산시스템, 컴퓨터네트워크, 이동컴퓨팅 등