

고성능 무중단 TCAP의 설계 및 구현

임옥희⁰, 박희숙, 이남석, 이상호

충북대학교 전자계산학과

(Imbung@locusnet.com, hspark@comingtek.com, chxyz@chungbuk.ac.kr, shlee@chungbuk.ac.kr)

Design and Implementation of a High Quality Dual TCAP

Ok-Hi Im⁰, Hee-Suk Park, Nam-Suk Lee, Sang-Ho Lee

Department of Computer Science

Graduated School, Chungbuk National University

요 약

오늘날 우리는 많은 발전을 거듭한 지능망 서비스를 일상 생활에서 직간접적으로 접하고 있으며, 서비스가 다양화 되고 그 품질이 중요시 되면서 TCAP(Transaction Capability Application Part)의 높은 성능과 안정성에 대한 요구가 증대되어 왔다. 본 논문에서 제시하는 고성능 무중단 TCAP 모델은 원격지에 위치한 다중 TCU(Transaction Capability User)에게 TCP/IP 소켓을 제공하고 폴링(Polling)을 수행하여 TCU상태를 효과적으로 관리하며, 네트워크 장애나 하위 스택 프로세스 문제로 서비스가 지속될 수 없는 경우 이중화 메커니즘을 적용하여 서비스의 중단을 최소화 한다. TCU 어플리케이션 개발자를 위해 이중화에 대한 고려 없이 TCAP 메시지를 사용할 수 있도록 API를 제공하여 실제 서비스 적용에 범용성과 편리성을 제공한다. 기존에 개발되어 사용되고 있는 TCAP보다 메시지 처리 및 전송 효율을 높이기 위한 프로세스 구조와 무중단 서비스 제공을 위한 이중화 모델을 설계하고 구현하였다. 설계 및 구현 내용을 검증하기 위해 ITU-T에서 권고하는 적합성 시험을 실시하고, 테스트 베드를 구축 해 성능 시험을 수행한다.

1. 서 론

1980년대 중반을 기점으로 통신망이 정비되고 전화보급이 확대됨에 따라 회선 접속서비스 외에 각종 부가서비스의 요구가 증대되었다. 이 같은 요구를 충족시키기 위해 교환기와는 별도로 SCP(Service Control Point)와 SMS(Service Management System)를 집중 구성하고 공통선 신호방식(CCS7, Common Control Signaling No.7)의 고속, 다중, 다양한 신호기능을 활용 함으로써 여러 가지 서비스를 신속하고 경제적으로 제공하는 지능망 서비스가 개발 되기 시작했다. [1][2]

공통선 신호방식에서 TCAP(Transaction Capability Application Part)은 부가 서비스에 있어서 망 서비스 제어국(SCP/SMS)에 대한 SLP(Service Logic Program)데이터나 가입자 정보 액세스 같은 비교적 데이터 양이 적고 직시성이 요구되는 SSP-SSP, SSP-SCP, SCP-SCP간에 범용성 신호 방식을 제공한다.[2]

지능망 서비스가 다양화 되고 그 품질이 중요시 되면서 TCAP(Transaction Capability Application Part)의 높은 성능과 안정성에 대한 요구가 증대되어 왔다. 본 논문에서 제시하는 고성능 무중단 TCAP 모델은 원격지에 위치한 다중 TCU(Transaction Capability User)에게 TCP/IP 소켓을 제공하고 폴링(Polling)을 수행하여 TCU상태를 효과적으로 관리하며, 네트워크 장애나 하위 스택 프로세스 문제로 서비스가 지속될 수 없는 경우 이중화 메커니즘을 적용하여 서비스의 중단을 최소화 한다. TCU 어플리케이션 개발자를 위해 이중화에 대한 고려 없이 TCAP 메시지를 사용할 수 있도록 API를 제공하여 실제 서비스 적용에 범용성과 편리성을 제공한다. 기존에 개발되어 사용되고 있는 TCAP보다 메시지 처리 및 전송 효율을 높이기 위한 프로세스 구조와 무중단 서비스 제공

을 위한 이중화 모델을 설계하고 구현하였다. 설계 및 구현 내용을 검증하기 위해 ITU-T에서 권고하는 적합성 시험을 실시하고, 테스트 베드를 구축 해 성능 시험을 수행한다.

본 논문의 구성은 제2장에서 TCAP의 구조를 설명하고, 제3장에서는 TCAP의 효율을 높이고 무중단 서비스를 위한 고성능 무중단 TCAP 모델을 설계한다. 제4장에서는 설계한 모델에 대한 구현 및 적합성 시험 결과와 테스트 베드를 구축하여 수행한 성능시험 결과를 분석하고, 마지막으로 제5장에서 결론을 맺는다.

2. TCAP 프로세스 구성

TCAP은 (그림1)과 같이 2개의 서브 레이어로 구성되어 있으며, 서브레이어 내 DHA, CHA, THA 프로세스는 각각의 프로세스간 통신을 위하여 독립적인 프리미티브를 갖는다.

2.1 CSL

CSL은 TCU로부터 상대 노드에 대한 처리 요구 오퍼레이션이나 결과값 지시자를 수신하여 컴포넌트 처리를 수행하는 CHA(Component Handling Area)와 TCU로부터 상대 노드까지의 논리적인 통신로를 설정, 해제를 위해 다이얼로그 처리를 수행하는 DHA(Dialogue Handling Area)로 구성된다.[1][3] CHA는 컴포넌트 인코딩과 디코딩을 수행하는 CCO(Component Coordinator)와 Invoke ID(IID)를 키 값으로 컴포넌트 상태를 관리하는 Invoke State-Machine(ISM)으로 구성된다(그림1참조).

2.2 TSL

CSL에서 생성된 하나의 다이얼로그에 대해 상대 노드로 전송하기 위한 논리적인 전송로를 확인하여 해당 요구

메시지를 송신하고 처리 지시 메시지를 수신한다.[1][3] TCAP 메시지의 인코딩과 디코딩을 수행하는 TCO(Transaction Coordinator)와 Transaction ID(TID)를 키 값으로 다이얼로그 상태를 관리하는 TSM(Transaction State-Machine)으로 구성된다(그림1 참조).

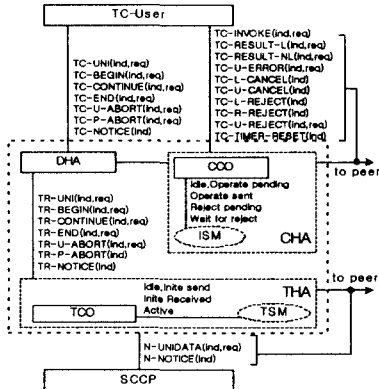


그림1 TCAP 내부 레이어 및 프리미티브 구조도

3. 고성능 무중단 TCAP 설계

3.1 고성능 구현을 위한 설계

3.1.1 CSL, TSL 부 통합

TCAP 내부 메시지 처리 절차를 최소화 하기 위해 (그림2)와 같이 CSL과 TSL 두 레이어를 하나의 레이어로 통합하고 TSM은 Dialogue ID를 키 값으로 DSM(Dialogue State-Machine)을 운영한다. CSL, TSL 간 프리미티브 인코딩·디코딩 단계를 축소하고 DSM을 운용하여 TCAP 내부 메시지 처리 절차를 간소화 한다.

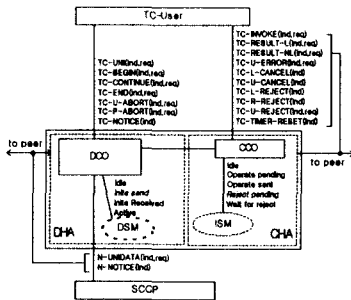


그림2 CSL, TSL 통합 설계

3.1.2 TCU socket과 IND socket의 분리

TCU와 TCAP간 송수신 메시지를 분류하면 (표1)과 같이 Local TCU의 요청 메시지와 Remote TCU의 처리 지시 메시지 등으로 분류가 되며, (그림3)과 같이 TCU와 IND socket을 두어 TCP/IP 메시지 트래픽을 분산시킨다.

(표 1) TCU-TCAP간 송수신 항목

Type	TCU -> TCAP	TCU <- TCAP
	DID Request	메시지 수신 ACK
	Component Request	DID return
	Dialogue Request	Indication Message
	Polling Response	Polling Request
	Status Request	Status Response

3.1.2 TCU, TCAP 간 폴링 방식

TCAP은 시스템 환경이나 서비스 특성에 따른 타이머 값을

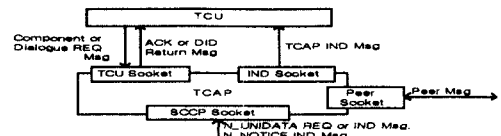


그림3 메시지 송수신 트래픽 분산을 위한 소켓구조
 기준으로 TCU와 물리적인 연결 상태를 폴링 방식을 통해 주 기적으로 검사하며, TCU의 상태 변경 이벤트 발생 시 상태 값(Fault, Reactive, Active)을 Shard memory에 저장하고 SCCP와 Peer TCAP으로 메시지를 전송함으로써 최신 망 정보를 서로 공유하게 된다.

3.2 무중단 서비스를 위한 이중화 설계

3.2.1 Active & Standby 모드

TCU <-> TCAP <-> SCCP 간의 메시지 송수신은 Active TCAP만이 수행하며, Standby TCAP은 Active TCAP으로부터 TCU/SCCP에서 수신 된 메시지를 Peer socket으로 동일하게 수신하여 처리함으로써 다이얼로그나 컴포넌트 정보, Shard memory 정보, 타이머 정보 등을 Active TCAP과 동일하게 유지하여 절체 발생시 호 손실을 최소화한다(그림4).

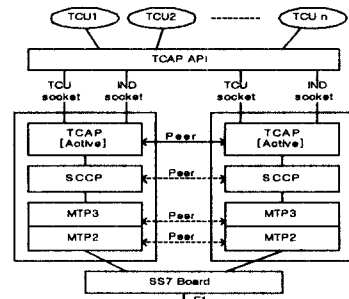


그림4 TCAP 이중화 구조도

3.2.1 절체 발생 시 처리 메커니즘

서비스 절체가 발생하는 경우를 하위 스택 오류나 네트워크 장애로 인한 절체와 TCU, TCAP간 네트워크 상태 불량으로 인한 절체로 구분하였다.

3.2.1.1 하위 스택 오류나 네트워크 장애로 인한 절체

하위 스택 오류나 네트워크 상태 불량으로 더 이상 서비스가 불가능할 때 SCCP는 Active TCAP으로 서비스 절체 이벤트를 송신하고, Active TCAP은 Standby TCAP에게 절체 사실을 통보하게 된다. Standby TCAP은 TCAP 상태 값을 Active로 변경 후 동일 하게 유지되던 호 관련정보를 기반으로 서비스를 제공하게 된다. (그림5)의 절차와 같이 지정된 메커니즘을 수행함으로써 Standby TCAP에서 바로 서비스가 진행되며, 호 손실 율을 최소화한 서비스를 제공할 수 있다.

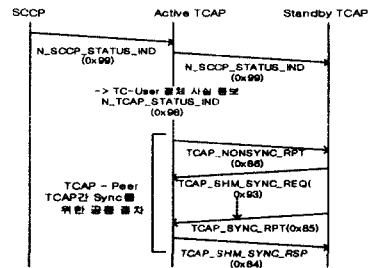


그림5 SCCP->TCAP 서비스 절체 시나리오

3.2.1.2 TCU-TCAP 네트워크 상태 불량으로 인한 절제

폴링을 수행한 결과 TCU, TCAP 간 연결 상태가 모두 'Fault'이면, 이를 TCAP이 감지하여 절제 이벤트를 수행하게 된다. Standby TCAP의 TCU 상태와 Active TCAP의 TCU 연결상태를 비교해 서비스 절제를 수행하고 이를 SCCP와 Standby TCAP으로 통보한다. 절제 되었던 TCAP의 활성화 시 (그림5)의 'TCAP-Peer TCAP간 Sync를 위한 공통절차'가 동일하게 수행되어 Peer간 동일한 정보를 유지한다.

4. 구현 및 성능 분석

4.1 구현

4.1.1 구현 환경

TCAP을 구현하는 하드웨어 및 소프트웨어 사양은 COMPAQ DS20 system, SS7 board(2 port)를 장착했으며, 운영체제는 UNIX 4.0F(Digital 64), 언어는 C를 이용하여 구현하였다.

4.1.2 프로세스 구성

개념적인 전체 소프트웨어 블록을 나타내면 (그림6)과 같이 기능별 프로세스와 공통정보를 포함하는 Shard memory, Timer 등으로 구성되며, 다이얼로그와 컴포넌트의 정상적인 경우와 비정상적인 경우에 대한 통계 정보와 API를 제공한다.

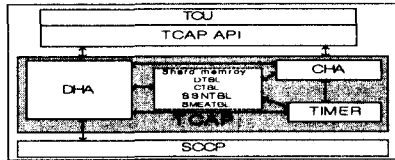


그림6 TCAP의 소프트웨어 블록도

4.2 성능분석

4.2.1 적합성 시험

적합성 시험은 (그림7)과 같이 DEC시스템에 개발한 TCAP 프로세스를 구동한다. SCCP는 DEC SS7에서 제공하는 모듈의 SCCP API를 이용했으며, TCU는 INAP을 이용하여 시험 시스템을 구성한다. K1297에서 ITU-T Q.787[8]을 FORTH 프로그래밍하여 권고하는 137개 항목의 적합성 시험을 수행한 결과는 (표2)와 같다.

(표2) 적합성 시험 결과

테스트 내용	항목	테스트 외 항목
Transaction Sublayer (TSL)	64	137 (16 (DPC 오류 등))
Component Sublayer (CSL)	73	

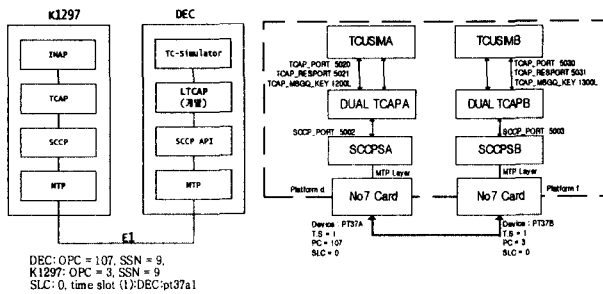


그림7 기능 시험을 위한 구성도 그림8 성능시험 테스트 베드

4.2.2 성능시험

TCAP의 성능 시험 테스트 베드는 (그림8)과 같이 하나의 DEC 시스템에 두 개의 SS7 board를 장착 하여, 메시지 송수신 절차를 지정 횟수만큼 반복적으로 수행하고 시작 시간과 종료시간을 체크 해 처리시간을 산출 하였다.

시뮬레이터는 INAP을 모델로 TCAP에서 제공하는 API를 이용해 개발 하였다.

초당 평균적으로 640개의 다이얼로그를 처리하며, SS7 Stack 전체 처리 효율(0.8 erl)로 볼 때 높은 성능을 보인다.

5. 결론

본 논문에서 제안하는 고성능 무중단 TCAP은 높은 성능과 이식성을 위해 ITU-T 인터페이스를 지원하고, 네트워크나 스택 상태에 상관없이 최대한 서비스 중단이 없도록 절제 메커니즘과 메시지 체계를 두었다

앞으로 좀 더 고려되어야 할 사항은 절제 되었던 시스템의 재 활성화 시 다이얼로그와 컴포넌트 정보, Shard memory 정보, 타이머 정보 등을 Active TCAP에서 Standby TCAP으로 TCP/IP 소켓 메시지를 이용해 전송하도록 구현되어 있다는 점이다. 이는 전송 데이터 양이 적을 때는 효과적이고 경제적이거나 데이터 양의 증가 할수록 효율이 감소하여 서비스 절제 시 전체적인 성능 저하를 가져 올 수 있다. 이는 추가적인 메모리 디바이스 도입과 그에 따른 인터페이스 개발 등으로 개선이 이뤄 질 수 있다고 보며, 장비 부족으로 인해 이종화 테스트 베드에서 성능 시험을 수행하지 못하였기 때문에 추가적인 테스트와 성능 분석이 필요하다.

TCU 시뮬레이터가 아닌 실제 서비스 사이트에 적용해 INAP이나 IS41등과 연동하여 발생할 수 있는 예외 사항에 대한 검증이 필요하며, 이러한 검증이 이뤄진다면 실제 프로토콜 스택에 적용되어 안정적인 서비스를 제공할 수 있다고 본다.

6. 참고문헌

- [1] 한국전자통신연구원, "속, 알기 쉬운 공통선 신호방식", 한국전자통신연구원, 1994. 1.
- [2] 최고봉, 김기령, 김태일, 윤병남, "지능망 기술", 한국통신학회, 홍릉과학출판, 1996. 10
- [3] ITU-T Q.771: "Specifications of Signaling System No.7; Functional Description of transaction capabilities", 1997
- [4] ITU-T Q.772: "Specifications of Signaling System No.7; Transaction capabilities information element definitions", 1997
- [5] ITU-T Q.773: "Specifications of Signaling System No.7; Transaction capabilities formats and encoding", 1997
- [6] ITU-T Q.774: "Specifications of Signaling System No.7; Transaction capabilities procedures", 1997
- [7] ITU-T Q.775: "Specifications of Signaling System No.7; Guide-lines for using transaction capabilities", 1997
- [8] ITU-T Q.787: "Specifications of Signaling System No.7; Transaction capabilities Test specification", 1993
- [9] 도경민, 최고봉, 이형호, "TDX-10 SSP 개발을 위한 TCAP 기능 구현에 관한 연구", JCCI91, 보은, 1991, 4.27, p.217-220