

교환기에서의 국번 번역 분산 처리 방안

이 정 옥, 김 덕 중, 범 순 균, 조 시 철
LG정보통신(주) 중앙연구소 교환연구단

Distributed Number Translation Processing By Prefix Translation Library

Jeong-Ok Lee, Deuk-Jung Kim, Sun-Kun Beom, Si-Cheol Cho
Switching Research, R&D Center, LGIC. Ltd

요 약

종래 국내 교환기 시스템에서 번호 번역 처리 등의 공통 절차들은 전통적으로 중앙 프로세서에서 수행한다는 개념에서 벗어나지 못했다. 이들 호처리 수행의 공통 절차들은 최번시 중앙 프로세서로 상당한 부하를 가중시키고, 곧 시스템의 전체 호처리 수행 능력이 중앙 프로세서의 성능에 종속되는 문제점으로 지적되어 왔다. 본 고는 일반 국번 번역 기능을 전담하는 신규 라이브러리를 ASS 서브시스템에 구성하므로서 시그널 송수신 절차를 간소화하고, 국번이나 착신번 번역 기능이 중앙 집중형 프로세서에서 제어되므로서 갖는 시스템 성능의 제약을 해결하기 위한 방안을 제시하고자 한다.

I. 서론

프로세서간 통신 처리 속도는 시스템 성능에 절대적인 영향을 준다. 이에 따라 교환기 H/W적인 면에서는 프로세서간 통신 경로 구조를 단순하면서도 병목현상이 없는 속도를 구현하여 충분한 성능을 내도록 꾸준히 개선해 왔으며, S/W적인 시퀀스상에서도 불필요한 IPC(Inter Processor Communication)가 발생하지 않도록 호처리 시퀀스 절차를 간소화하는 다양한 방안들이 모색되고 적용되어 왔다.

기존 TDX 시스템에서 번호 번역 기능은 중앙 프로세서에서 전체 시스템의 번호를 번역해 주는 중앙 집중화된 구조를 가지고 있었다. 호처리 절차 중 필수적인 번호 번역 처리가, 중앙 프로세서로의 부하를 집중시키므로서, 결과적으로 순간호 폭주 상황이나 최번시 중앙 프로세서의 최대

성능은 곧 시스템 전체 성능을 좌우하는 중요 변수이자 **Bottle-Neck**으로 지적되어 왔다. 다양한 신규 서비스들의 수용이 급속도로 확장되고 대용량화로 진화됨에 따라 이러한 단점은 더욱 크게 부각될 수 밖에 없었고 새로운 알고리즘 요구되었다.

호 접속의 Key가 되는 번호 번역은 일반적으로 국번부와 착신번부 번역으로 구분되는데, 본 고에서는 이 두개의 번역 처리 중 국번 번역 처리 부분을 중앙프로세서에서 분산된 제어 프로세서에 이관하므로서 프로세서간 IPC 송수신 절차를 간소화하고 보다 효율적인 호처리 구조와 성능 개선 방안을 제시하고자 한다.

II. 교환기에서의 번호 번역 처리 구조 소개

TDX계열 교환기는 일반적으로 물리적 자원들의 구성 위치에 따라 ASS(Access Switching Subsystem), INS(Interconnection Network Sub-system),

CCS(Central Control Sub-system)로 구성되며 원격 가입자 수용을 위해 Remote ASS(RASM, SRASM) 등의 장치를 ASS에 수용한다.

ASS는 가입자 및 중계선 인터페이스, 타임스위치, 각종 신호 장치, 패킷 등을 구비하여 대부분의 호처리 기능과 자체 운용 및 유지 보수 기능 등을 수행하며 필요시 ASS 단위로 증가시킬 수 있으므로 시스템 적으로 수평 분산 구조를 가진다.

INS는 시스템의 중심에 위치하여 ASS 상호간 혹은 ASS와 CCS 사이를 연결해 주는 기능을 수행한다. 번호 번역과 라우팅 및 스위칭 기능과 같은 호처리 공통 기능들을 수행하는 블록들은 INS에 위치한다.

CCS는 시스템 총괄적인 운용 및 유지보수 기능을 수행하는 서브 시스템으로 시스템 차원의 유지보수 및 운용 관리 통계 기능, MT나 DISK의 제어 관리, 운용자 명령어 제어 기능, TMN(Telecommunication Management Network) 등 타 시스템과의 연동 기능을 수행한다.

호 성립에서부터 해제까지의 호 처리 기능은 ASS에 위치한 호 제어 블록들과 INS에 위치한 번호 번역 블록, 스위치 블록 및 루팅 블록들의 연계에 의해 이루어 진다. 번호란 발신 가입자가 원하는 착신지를 찾아가는 Key이다. 교환기 시스템은 자국내에서 중계 또는 착신 처리가 가능한 국번과 특번 및 착신번 그리고 지역번호에 대한 분석을 위해 트리 구조의 테이블 형태로 이들 데이터를 저장하며 이는 시스템의 중요한 국데이터 중의 하나로 분류된다.

국번호는 12진 트리 구조를 갖는 11개의 평면으로 구성되며, 0번 번호 평면은 시외호 국번(toll_call), 가입자 특수 서비스 국번(*로 시작되는 국번들)과 같이 시스템에서 유일하게 존재하

는 국번호가 배치되고, 1~10번까지의 번호 평면은 지역번호와 각각 연계되어 있어, 지역번호에 귀속되는 교환기 내부호 국번(internal_call), 시내호 국번(local_area_call)과 특번(119,112 등)이 배치된다.

착신번은 10진 트리 구조를 갖는 번호 평면에 저장하게 되며, 각 교환기 내부 국번에 하나의 착신 번호 평면을 할당하여 운용하게 된다. 자국 가입자 국번마다 하나씩 착신 번호 평면을 할당하기 때문에 착신 번호 평면은 유동적인 다평면 구조를 가질 수 밖에 없는 것이다.

2.1 기존 교환기에서의 국번호 번역 알고리즘

번호 번역 블록은 번역 요구 받은 디지털을 Key로 국번 및 착신번 테이블을 포인트-투-포인트 방식으로 탐색하여 일치된 국번 또는 착신번호가 존재할 경우 해당 정보의 결과값을 호 제어 블록으로 통보하는 기능을 담당한다.

국번 및 착신번 관련 데이터를 저장하는 테이블은 개별적인 ASS에서는 접근이 불가능하고 중앙 프로세서만 액세스가 가능하다. 번호데이터처리 블록은 중앙프로세서내의 번호 관련 국 데이터를 생성/참조/변경/삭제하는. 운용자 명령어 수행 시 데이터 핸들링을 담당한다.

그림 1.은 가장 단순한 형태의 번호 번역 절차를 나타낸다.

가) 발신 호처리 블록은 국번 번역을 위한 일정 개수만큼 디지털을 수신하면 번호 번역 블록으로 국번 번역을 요구한다.

나) 번호 번역 블록은 국번 테이블을 탐색하여 일치하는 국번이 있는지 조사한다.

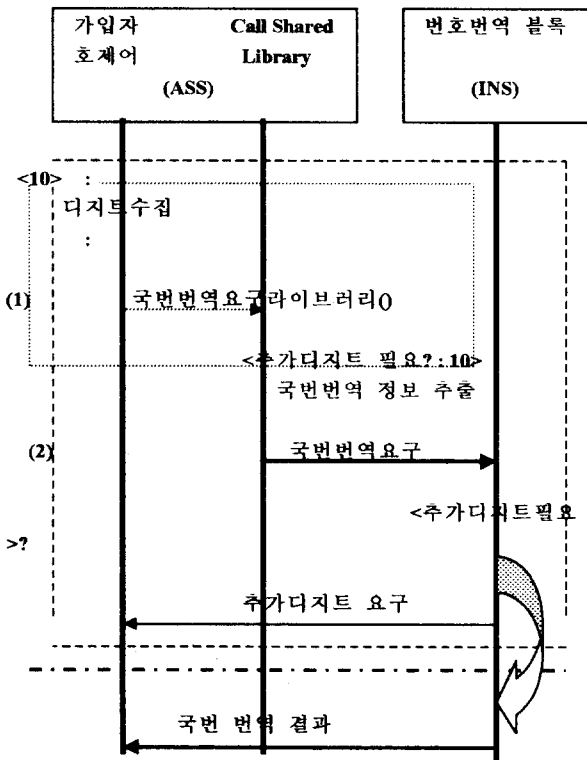


그림 1. 종래 기술에서의 국번 번역 시퀀스
 : 내/외부 IPC
 → 프로시저 호출

- 다) 수신한 국번을 번역하여 정상적인 국번이면 해당 국번이 갖는 호의 성격에 따라 출중계호, 입중계호, Transit 호 또는 자국호 처리에 필요한 정보를 국번 정보 테이블에서 구하여 국번 번역 결과 IPC를 송신하고, 출/중계호일 경우 루팅에 필요한 디지털이 충분하면 루팅블록으로 출중계 루팅을 요구한다.
- 라) 국번 번역 결과 수신한 디지털이 부족하면 국번 번역에 필요한 디지털 갯수를 구하여 발신 측으로 추가 디지털을 요구하고, 기타 국번 결번 또는 번호 변경의 경우에 해당 결과를 송신한다.
- 마) 추가 디지털 요구를 수신한 발신 호처리 블록은 대국 또는 단말로부터 필요한 만큼의 추가 디지털을 수집한 후 다시 번호 번역 블록으로 국번 번역을 요구한다

이 그림에서 볼 수 있듯이 국번 번역 절차에 필요한 ASS와 INS간 외부 메시지는 best case의 경우 2개(번역 요구-결과)가 필요하나, 번역에 필요한 디지털이 충분하지 못할 경우의 worst case를 (점선(2)의 box를 반복 수행할 경우) 반복할수록 발생하는 메시지의 개수는 2배로 증가됨을 알 수 있다.

이는 최번시에 병목 현상을 가중시켜 호 지연 시간이 길어지며, 번호 번역을 요구한 모든 호들이 대기 상태로 있으면서 시스템 자원들을 차지한 상태로 남아있게 되는 심각한 성능 저하의 요인을 제공한다.

III. 국번 번역 라이브러리를 이용한 분산 방안

2장에서 본 바와 같이 종래의 시스템에서는 모든 번호 번역 처리가 중앙프로세서에 의존하므로 호 과다 발생시 중앙 프로세서의 부하가 집중되는 문제점을 안고 있었다. 새로 제안하는 시스템에서의 번호 번역 기능은 국번부 번역과 착신번부 번역의 2단계로 나누어 이중 일반 국번 번역 처리만을 담당하는 블록을 신설하고, 중앙서브 시스템에 위치하는 번호 번역 블록은 특번이나 착신번에 대한 번역 처리만을 담당하도록 한다. 불필요한 내부 메시지 발생을 해결하기 위해 ASS내 특정 공통 영역에 존재하는 국번 번역 라이브러리 블록으로 구성하면 내부 또는 외부 IPC 송수신 절차없이도 프로시저의 호출과 결과 값 반환의 형태로 국번 번역 처리가 가능하다.

이러한 동작 실현을 위해서는 먼저 국번 관련 테이블은 INS 뿐만 아니라 모든 ASS에서도 액세스가 가능한 중복 데이터 테이블이어야 하며, 이에 따라 번호 데이터 처리 블록의 데이터 처리 절차를 변경하고, 국번 관련 중복 데이터들은 DataBase 감사 시 최우선 순위의 감사를 수행하

여 불일치가 발생되지 않도록 방지하며, 불일치 발생시 자동 복구되도록 관리한다.

국번 번역 라이브러리 블록은 프로시저 호출시 입력 파라미터로 넘겨진 입력 파라미터중 디지털을 Key로 국번 테이블을 직접 탐색하고 일치된 정보가 존재할 경우 결과값을 출력 파라미터로 실어 반환하는 번역 기능을 수행하는 프로시저들의 집합체이다.

각 호제어 블록들이 마치 자기 내부의 프로시저를 호출하는 것처럼 국번 라이브러리에 프로시저로 호출할 수 있도록, 모든 호제어 블록들이 접근 가능한 미리 정의된 공통 영역에 라이브러리 블록이 로딩되고 링크되도록 구성한다.

또한 국번 번역 라이브러리와 이를 사용하는 호제어 블록간의 인터페이스 절차를 위한 인터페이스 라이브러리 블록이 새로 생성되어야 한다. 이 블록은 국번 번역 라이브러리 블록 내부의 각 프로시저들의 Logical Address 를 Copy 하여 저장하고 있으면서 호제어 블록에서 국번 번역 프로시저 호출시 해당 프로시저의 실제 Address 로 1:1 대응시키는 매개 역할을 담당한다. 국번 번역 라이브러리 블록을 사용하고자 하는 블록은 실행파일 생성 단계에서 인터페이스 라이브러리의 실행 파일인 archive 화일을 Link 시키고 초기화 하여야 한다.

3.1 알고리즘

호제어 블록 및 Call Shared Library와 국번 번역 라이브러리는 서로 미리 약속된 국번 번역 라이브러리 호출 패턴과 반환 패턴에 따라 파라미터를 주고 받음으로서 인터페이스한다.

그림 2.는 국번 번역 라이브러리 블록을 통한 국번 번역 시퀀스를 보여준다.

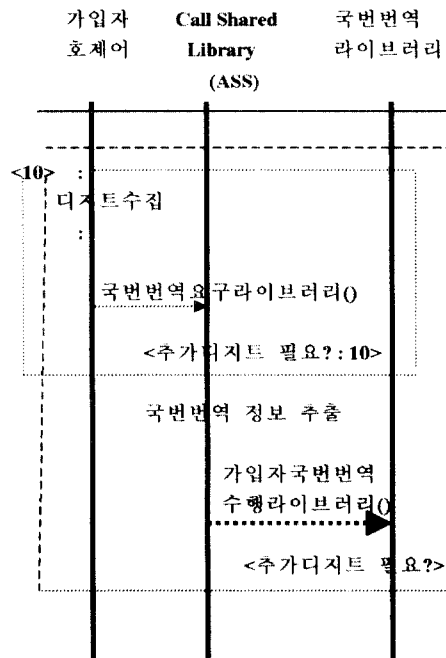


그림 2. 라이브러리를 통한 국번 번역 시퀀스

가) 국번 번역에 필요한 디지털이 수집되면, 호제어 블록은 CSL(Call Shared Library :호제어 공통 프로시저의 집합)에 있는 국번 번역 요구 라이브러리(pd_PfxTranSUB_NTL)를 통해 국번 번역에 필요한 각종 정보를 추출하고 이 정보를 파라미터로 국번번역라이브러리를 호출한다. 국번 번역 프로시저 호출시 국번 번역에 필요한 정보를 개별 입력 파라미터로 전달하기에는 너무 많은 정보이므로, 특정 위치에 저장된 호 정보의 위치 pointer를 파라미터로 전달한다. 이 정보는 국번 번역 라이브러리가 직접 액세스 할 수 있는 위치에 존재하며 입력 정보로 전달된 호 정보 위치 포인터를 인덱스로 호 정보를 추출하여 국번 번역에 필요한 값들을 얻어낸다.

호출 결과 반환 value로는 '국번번역 수행 여부'를 알려주는 파라미터와, 수행 완료시 상세 정보를 실은 '국번번역 결과' 라는 두가지의 파라미터가 존재한다. '국번 번역 수행 여부' 파라미터에는 '자체번역완료' / '추가정보필요' / '중앙번

호번역 블록으로의 재전달' 이라는 3가지 패턴이 존재한다.

나) 국번 번역 프로시저는 전달된 호정보 내용을 토대로 디지털을 Key로 국번 테이블을 탐색하여 일치된 정보가 존재하는지를 조사한다.

번역 요구된 해당 국번이 특번인 경우 자체 라이브러리내에서 번역이 불가능하므로, 이는 다시 INS에 존재하는 번호 번역 블록으로 '국번 번역 요구' IPC를 이용하여 재 요청하고 '수행 여부' 결과값으로서 '중앙번호번역 블록으로의 재전달'을 반환한다. 호 제어 블록은 이 값을 받으면 중앙의 번호 번역 블록으로부터 결과를 기다린다.

번역 요구된 국번의 정보가 번역에 필요한 디지털만큼 충분하지 않을 경우, '수행 여부' 결과값으로서 '추가 정보필요'를 반환하고 '국번 번역 결과' 파라미터에는 '추가 디지털 요구'를 반환한다. 호 제어 블록은 이 정보를 받으면 '국번 번역 결과' 에 실린 추가 디지털 만큼의 디지털을 더 수집한 뒤 국번 번역 요구 라이브러리 호출을 반복한다.

국번 번역 요구된 호의 디지털 정보도 충분하고 해당 국번이 특번이 아닐 경우 '수행 여부' 반환값엔 '자체번역완료'를 신고 '국번번역 결과' 값에는 국번 정보에 따라 자국호 라든가 입중계호 또는 출중계호 및 중계호 등의 결과값이 반환된다.

가입자 또는 중계선 제어 블록들은 라이브러리 블록내 프로시저를 호출하므로써 국번 번역을 수행하고 리턴값에 국번 번역 결과를 반환한다.

다) INS에 존재하는 번호번역 블록은 국번 번역 라이브러리에서 번역이 불가능한 모든 번호번역에 대한 처리를 수행한다. 즉 가입자 및 입중계호에 의한 특번 및 착신번 번역 요구를 수신하여 특번 번역 및 착신번 번역 결과, 발신측으

로는 번호 번역 결과를 발신 측으로 보내고 자국 착신번인 경우 착신점유 요구를 보내는 역할을 담당한다.

IV. 결 론

과거 기본 호처리 수행시 국번 번역을 위해 ASS-INS간 최소 2개 이상의 외부 메시지가 필요하였으나 새로운 시스템에서는 일반 국번 번역을 위한 내/외부 메시지는 불필요하게 되었으며, 기본 호처리 수행중 번호 번역을 위해 존재하는 ASS-INS간 IPC는 자국호의 경우 "착신번 번호 번역 요구(ASS->INS)"과 "착신 점유 요구(INS->ASS)"만으로 가능하며, 중계호의 경우 "라우팅 요구(ASS->INS)" 만이 필요하다. 또한 그에 따른 Control블럭의 Stack Size도 최소로 하여 Memory를 최적화하고 호 지연 시간을 줄일 수 있을 뿐만 아니라 각 프로세서의 리소스 점유 시간을 줄일 수 있고 스위칭 시간을 극소화 시킬 수 있게 되었다.

실제 본 알고리즘을 자사교환기에 구현하여 성능 시험을 한 결과 종래 알고리즘에 비해 INS의 최대 성능이 증대하였으며 모든 호처리상에서 문제가 없음이 증명되었다.

이로써, 과거 TDX 계열 시스템에서 관습적으로 이루어졌던 집중 번호 번역 처리 구조를 탈피하고, 불확실한 트래픽 환경에서 보다 안정적이고 경제적인 S/W 구조로 진화하는 계기를 마련하였다.

참고 문헌

- [1] "TDX-10 개량형 교환기"-시스템 개요, LG정보통신
- [2] "TDX-100 전자 교환기 기술 요구서", 한국통신