

# (N+1) 형식의 지능망 서비스 시스템에서의 부하균형 분석

노용덕\*, 김종용\*, 송상훈\*, 심장섭\*\*, 서인수\*\*

\* 세종대학교 컴퓨터공학과 교수

\*\* 데이콤 종합연구소

## Load Balancing Analysis For (N+1) Type Intelligent Network Service System

Noh Yongdeok\*, Kim Jongyong\*, Song Sanghoon\*, Shim Jangsup\*\*, Suh Insoo\*\*

\* Sejong University, Dept. of Computer Science

\*\* Dacom Research & Development Center

### Abstract

지능망은 소비자 또는 가입자의 다양한 욕구를 신속하고도 효과적으로 제공하며 동시에 새로운 서비스에 대한 수요가 있을 경우에 이에 쉽게 대처할 수 있는 차세대 통신망이다. FEP - BEP 구조를 갖는 (N+1) 형식의 지능망 서비스 시스템에서의 서비스에 대한 부하균형 방법으로 각 SCP-BEP 시스템별로 서비스의 종류에 따라서 부하를 할당하는 방식이 있으며, 이를 국내 통신회사에서 실제로 적용하고 있다. 이러한 방식의 장점은 부하균형을 동적으로 할당할 수 있고, 신규 서비스의 등록이나 기존 서비스의 갱신시에 기존에 제공하던 서비스에 대한 호에 미치는 영향을 최소화 할 수 있다는데 있다. 본 논문에서는 SCP-BEP 에 제공하는 서비스별로 할당하는 부하량에 따른 시스템의 성능추이를 시뮬레이션 기법에 의하여 추적하고 그 결과에 대하여 기술한다.

### 1. 서론.

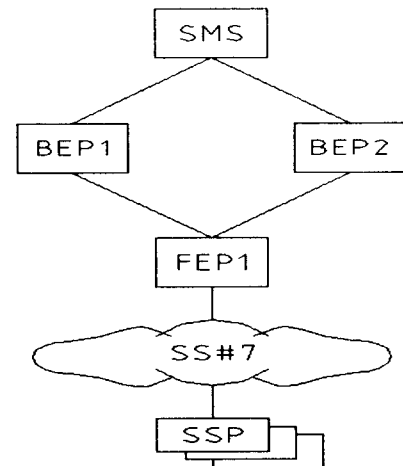
지능망은 소비자 또는 가입자의 다양한 욕구

를 신속하고도 효과적으로 제공하며 동시에 새로운 서비스에 대한 수요가 있을 경우에 이에 쉽게 대처할 수 있는 통신망이다. 지능망은 SSP

(Service Switching Point), SCP (Service Control Point), SMS (Service Management System), SCE (Service Creation Environment) 및 IP (Intelligent Peripheral) 로 구성되어 있다 [1]. 여기서 SSP 와 SCP 는 SS#7 네트워크로 연결되어 있고, SCP 와 SMS 는 데이터 통신 네트워크로 연결된다. 사용자는 SSP를 통하여 지능망에 연결되며, SSP 는 사용자의 지능망 서비스 요청 여부를 감지하고 SCP 에 지능망 서비스를 요청하며, SCP 로부터 받은 정보에 따라서 필요한 네트워크 관련 작업을 수행한다. SCP 에는 데이터베이스 및 SLP (Service Logic Program) 이 있다. SLP 에는 서비스를 위한 제어로직이 있으며 데이터베이스에는 서비스 제공에 필요한 정보가 저장되어 있다. SMS 는 지능망 관리를 위한 시스템으로 SCP 에 지능망 서비스에 필요한 데이터베이스를 다운로드하거나 서비스 가입자와 제공자간의 인터페이스를 제공하고, 여러 가지 측정된 데이터를 수집한다. SCE 는 SLP 개발에 필요한 여러 가지 툴을 포함한다. IP 는 서비스 이용자와 지능망간의 다양한 정보 교환을 가능하게 하는 것으로, 음성 데이터 수신, 음성의 송출, 안내방송, 등의 다양한 기능을 갖는다.

본 논문에서의 지능망 서비스 시스템은 (N+1) 형식으로 설계된 시스템으로, 서비스 제어장치인 SCP (Service Control Point) 가 SCP-FEP (SCP - Front End Processor) 와 SCP-BEP (SCP - Back End Processor) 의 기능적으로 분리된 구조를 갖고 있다. 여기서 SCP-FEP 는 SS#7 프로토콜 스택 등 신호망과의 인터페이스를 담당하는 기능과 지능망 응용 프로토콜 (INAP : Intelligent Network Application Protocol) 처리기능을 포함하는 지능망 인터페이스 처리부로 구성되며, SCP-BEP 는 지능망 호(Call)에 대한 서비스 제공을 위하여 서비스 처리기능을 수행하는 서비스 처리부로 이루어진다. SCP 의 기능을 위

와 같이 분리함으로써 인터페이스 처리기능과 서비스 처리기능을 독립적으로 운용할 수가 있으며, 따라서 서비스 처리 용량의 증가나 신규 서비스의 확대 실시에 따른 시스템의 확장을 각 기능별로 할 수가 있다[2]. 또한, 각각의 SCP-FEP 및 각각의 SCP-BEP 는 서로 독립적으로 동작하며, SCP-BEP 의 경우에 동일한 종류 또는 다른 종류의 서비스를 하나의 SCP-BEP 별로 제공한다. 지능망에서는 다양한 서비스의 제공으로 트래픽에서의 오버로드가 생길 수가 있으며, 따라서 수요자를 만족시키는 QoS를 제공하는 것은 매우 중요한 일이다. 이를 위하여 SCP-FEP 와 SCP-BEP 간의 부하균형 (Load Balancing) 에 대하여 논하고, 여러 가지 부하균형에 대한 경우를 시물레이션 기법을 사용하여 시험하고 그 결과를 정리하였다.



[그림 1] 지능망의 구조

## 2. SCP-FEP 와 SCP-BEP 간의 부하균형

1개의 SCP-FEP 와 2개의 SCP-BEP를 갖는 (N+1) 형식의 지능망 서비스 시스템을 [그림 1]에 정리하였다. SSP 에 전달된 지능망 호는

SS#7 신호망을 거쳐 SCP-FEP 로 전달된다. SSP 에서는 PC (Point Code)를 사용하여 SCP-FEP로 신호를 보낸다. SCP-FEP의 SS#7 Stack을 거쳐 INAP 처리 블록으로 전달된 호는 서비스 구분자 (예: InitialDP의 서비스키)로부터 처리가 요구된 서비스가 결정되고 INAP 처리 블록의 부하균형표에 있는 값에 따라 적절한 SCP-BEP로 전달된다. [표 1]은 INAP 처리 블록에서 관리하는 부하균형표를 나타낸다. 그리고, SCP-BEP에서는 전달된 지능망 호에 대해 요구되는 서비스 로직을 적용하여 서비스를 제공하게 된다.

[표. 1] 서비스별 부하균형표.

	BEP(1)	BEP(2)	...	BEP(n)
Service(1)	p(1,2)	p(1,2)	...	p(1,n)
Service(2)	p(2,1)	p(2,2)	...	...
...	...	...	...	...
Service(m)	p(m,1)	p(m,2)	...	p(m,n)

이러한 부하균형표는 각 SCP-FEP의 INAP 처리 블록마다 존재하며, INAP 처리 블록에서 각 신규 지능망 호가 SCP-FEP에 들어 올 때에, INAP 처리 블록에서는 그 호의 서비스 구분자로부터 서비스를 결정하고 부하균형표로부터 이 호가 처리될 SCP-BEP를 결정하는 것이다. 부하균형표에서 p(i,j)의 값이 0인 것은 서비스(i)가 SCP-BEP(j)로 전달되지 않을 것임을 나타낸다. 또한 모두 0의 값을 갖는 서비스의 경우는 서비스가 이 지능망 서비스 시스템에서 현재 제공되지 않음을 나타내는 것으로, INAP 처리 블록에서는 이러한 호에 대해서는 비정상호 처리 절차를 밟는다. 또한 부하균형표에 정의되지 않은 서비스 구분자를 갖는 신규 호에 대해서도 비정상호 처리절차를 밟는다.

SMP는 각 SCP-FEP 및 SCP-BEP로부터 정의된 과부하 점검요소들을 주기적으로 보고 받고 이로부터 적절한 성능 제어의 방법 및 수준을 결정한다.[2] 성능제어 방법으로는 SSP로부터 전달되는 호의 수를 제한하는 INAP 오퍼레이션 CallGap을 이용하는 방법 및 SS#7 Stack과의 연결 점 수를 이용한 부하균형 방법이 있다. CallGap 의 경우는 지능망 서비스 시스템 전체의 트래픽 과다 등과 같은 시스템 전체 성능 제어시에 사용되며, 부하균형의 경우는 어느 특정 SCP-BEP에 트래픽의 집중되는 경우에 호를 적절히 다른 SCP-BEP로 재분배 하기 위하여 사용한다. 그러나 여기서는 INAP 처리 블록이 SCP-FEP에 위치하므로 INAP 처리 블록은 지능망 호를 부하균형표에 따라 적절한 SCP-BEP로 전달하게 된다.

이 INAP 처리 블록의 부하균형 값은 SMS의 성능 관리 기능에 의해 자동적으로도 변경이 가능하며, 또한 허가된 서비스 운용자에 의한 수동 변경도 가능하다. 이러한 수동 제어에 의한 부하균형 값의 변경은 자동 제어에 의한 변경에 우선하는 것이 일반적이다. 이와 같은 지능망 시스템은 여러 종류의 서비스를 제공하며, SCP-FEP와 SCP-BEP 간의 부하균형 값을 할당하는 여러 가지 경우가 발생하므로, 이를 분석하기 위한 가장 좋은 방법은 시물레이션에 의한 것이다. 또한, 시물레이션 기법은 여러 가지의 시스템 구조와 그에 관계된 인자들에 대하여 조사할 수 있는 장점이 있다.

### 3. 시물레이션 모델 및 검증

부하균형을 위하여 윈도우 3.1하에서 SLAM SYSTEM 4.1[4]을 사용하여 시물레이션을 실시하였다. 모델에서는 각 호의 도착시간 및 처리시간이 지수분포를 따르는 것으로 가정하였다. 모

델의 검증을 위하여 상업용 지능망에서 시험호를 발생시킨 후에 얻은 결과와 시뮬레이션 모델을 실시한 후에 얻은 결과를 비교하였다. 상업용 지능망에서는 시험호를 발생한 후에 이에 대한 처리는 SCP-FEP1, SCP-BEP1, SCP-BEP2, 및 SMS를 사용하였으며, 40 CPS (Calls Per Second) 와 60 CPS 의 경우에 대하여 두 번씩 실시하였다. 여기서는 그 당시에 얻은 SCP-FEP, SCP-BEP, SMS 에 대한 CPU 사용량과 실행시간(Turnaround Time = 처리시간 + 대기시간)을 조사하고, 각각의 평균치를 시뮬레이션 모델의 결과와 비교하였다. [표 2] 및 [표 3]은 이때 사용한 SCP-FEP에서의 부하균형표와 모델 검증의 결과를 보여주고 있다. 특히, [표 2]에서 서비스 내의 숫자는 도착한 하나의 호가 요구하는 서비스의 확률 값을 나타낸다.

[표. 2] 부하균형표 : 경우1.

서비스	SCP-BEP1	SCP-BEP2
VPN (0.12)	0.5	0.5
AFS (0.10)	0.5	0.5
PN (0.35)	0.5	0.5
NP (0.43)	0.5	0.5

[표. 3] 모델의 동질성 검증

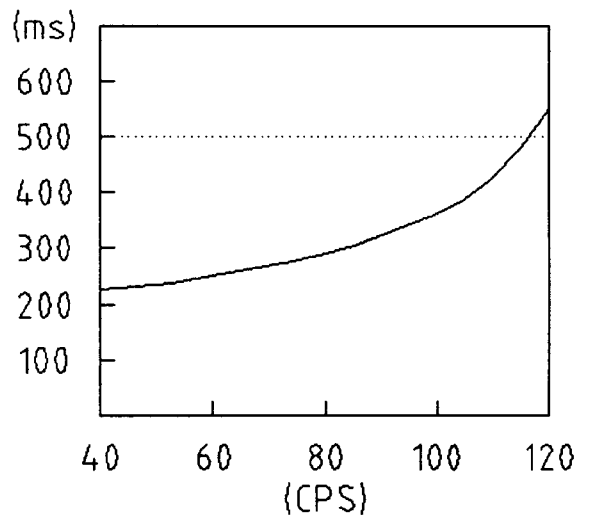
	SMS	BEP1	BEP2	FEP	Time
40 CPS	32	35	36	31	210
모델	33	39	38	32	227
Chi-Square 값 : 0.107 < 11.14					

	SMS	BEP1	BEP2	FEP	Time
60 CPS	43	50	46	35	318
모델	39	51	52	42	256
Chi-Square 값 : 4.820 < 11.14					

4. 모델의 분석

위의 모델을 사용하여 1개의 SCP-FEP 와 2개의 SCP-BEP, 및 1개의 SMS를 사용하는 환경 하에서 호의 수요가 증가함에 따라 반응하는 시스템의 성능에 대하여 살펴보고, 이에 대응하여 여러 가지 부하균형의 값을 적용하였다. 지능망에서는 지능망 시스템의 과부하에 대하여 여러 가지 기준을 정의하여 사용하고 있다. 여기서는 하나의 서비스 호가 SCP-FEP 에 들어온 후에 다시 연결을 끊는 신호를 보낼 때까지의 시간, 즉, 실행시간이 500 ms를 넘지 않게 하는 것을 목표로 삼는다.

먼저, [표 2]의 경우1에 대하여 40 CPS, 60 CPS, 80 CPS, 100 CPS, 및 120 CPS 로 호가 증가할 때에 실행시간의 추이를 조사하고 그 결과를 [그림 2]에 정리하였다. 그림에서 보듯이, 110 CPS 와 120 CPS 사이에서 실행시간이 500ms를 초과하고 있다. 따라서 서비스 처리를 위한 부하균형의 값에 변화를 주어야 한다. 대개 이러한 경우는 사용량이 적은 CPU 에 높은 부하를 할당하지만, 경우1에서는 들어오는 호를 SCP-BEP1 과 SCP-BEP2 에 균등히 배분하므로 CPU 의 사용량이 서로 근사하다. 따라서 [표 4]와 같은 부하균형표를 사용하였다.

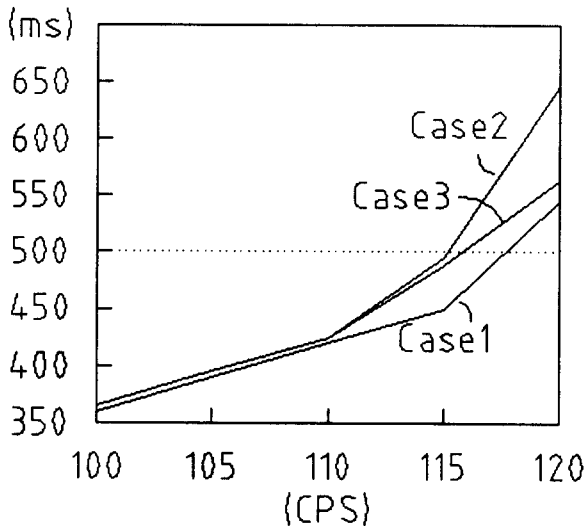


[그림 2] 각 호의 실행시간 추이.

[표. 4] 부하균형표 : 경우2, 경우3.

	경우2		경우3	
	BEP1	BEP2	BEP1	BEP2
VPN (0.12)	0.0	1.0	0.25	0.75
AFS (0.10)	1.0	0.0	0.75	0.25
PN (0.35)	0.0	1.0	0.25	0.75
NP (0.43)	1.0	0.0	0.75	0.25

[표. 4]에서 경우1과 경우2는 서로 극단적인 부하균형값을 갖고 있고, 경우3은 그 중간값을 할당한 경우이다. 이러한 부하균형값에 대하여 100 CPS에서 120 CPS까지 5 CPS 단위로 5분간 시뮬레이션한 결과를 [그림 3]에 정리하였다.



[그림 3] 각 경우에 대한 실행시간 추이.

[그림 3] 에서 보듯이, 경우1의 부하균형이 여전히 경우2나 경우3보다 실행시간이 적게 걸리는 것으로 보인다. 경우3에서는 그 결과가 경우1과 경우2의 중간에 위치하는데, 생각보다 경우1의 결과에 가까운 것은 시뮬레이션 모델에서의 모델링 방식에 기인한 것으로 보인다. 또한, 125 CPS 에서는 각 경우마다 실행시간이 급격히 늘어나

서, 각각 606, 674, 824ms 의 결과를 얻었다. 결론적으로, 현재 요청하는 서비스의 분포하에서는 경우1의 부하균형표가 보다 합리적이라고 결론질 수 있다.

### 5. 요약

여기서는 (N+1) 형태의 지능망 서비스 시스템 하에서의 부하균형 방식과 부하균형표의 값에 따라서 실행시간이 어떻게 변하는 지에 대하여 살펴 보았다. 그러나, 앞에서 지적하였듯이, 지능망 시스템의 부하를 측정하는 방식에는 실행시간이 외에도 CPU 시간의 사용률이나, 스택에서의 대기시간, 대기열의 크기, 하나의 SSP 에 대한 동시 트랜잭션의 수, 등이 판단기준이 된다. 그러나 여기서는 단순히 실행시간만을 대상으로 시뮬레이션을 실시하였다. 부하균형표를 작성하는 경우에도 매우 많은 경우의 수가 존재하므로 각 서비스별로 소요되는 CPU 시간에 따라서 부하균형값을 계산하고 그에 따른 시뮬레이션을 시도할 필요가 있다. 또한 아직 제공하지 않은 서비스가 실제로 사용될 경우에 대비하여 시스템에게 미치는 영향을 사전에 실험하는 것도 중요한 일이 될 것이다.

그러나 이러한 작업들은 시뮬레이션의 모델링에 있어서 실제 시스템 동작에 어느 정도 자세히 근접하게 설계했는 지에 따라서 결과에 커다란 영향을 미칠 수가 있다. 이 모델에서는 실제 데이터 수집의 어려움으로 시스템내의 프로세스 하나하나에 대하여 모델링을 하지 않았으므로, 이에 대하여 보다 폭 넓은 자료 수집과 모델링 작업을 수행하여야 할 것으로 판단된다.

### 참고문헌

1. Thorner, "Intelligent Networks", Artech

House, 1994

2. 김재혁, 김종주, 서인수, "지능망 서비스 시스템에서의 Load Balancing 기능 적용", 제4회 지능망 학술대회 논문집, 1999
3. 김종주, 송창환, 김재혁, 강용구, "지능망 서비스 시스템의 과부하 제어 고찰", AIN'97, 1997
4. Pritsker, "SLAMSYSTEM", Pritsker Corporation, 1990