

지능망에서의 이동 에이전트를 이용한 과부하 제어 메커니즘에 관한 연구

정희원 이광현*, 박승균**, 박주희**, 오영환**

A Study on an Overload Control Scheme for Intelligent Networks Based on Mobile Agent Technology

Kwang-hyun Lee*, Seung-kyun Park**, Ju-hi Park**, Young-hwan Oh** *Regular Members*

요약

전통적인 지능망 부하제어에서 개별적인 SCP(Service Control Point) 보호가 주요 관점이었다면, 본 논문에서는 지능망 전체 네트워크 자원 활용의 최적화를 통한 부하제어를 주요 관점으로 하였다. 이는 이동 에이전트 기술의 이용으로 가능해 지는데 이동 에이전트는 실행환경이 구현된 모든 시스템으로의 이동이 가능하고 독자적으로 작업 수행이 가능한 소프트웨어이다. 본 논문에서는 이동 에이전트를 이용한 과부하 제어 메커니즘과 이동절차를 제안하였다. 제안한 이동 에이전트를 이용한 과부하 제어 메커니즘은 이동 에이전트를 이용해 서비스 특성에 따른 지능망 과부하를 제어할 수 있고 이동 절차는 지능망 부하 변화에 따른 이동 에이전트의 이동 방법이다. 시뮬레이션결과 제안한 메커니즘은 기존 Call Gapping 메커니즘보다 SCP 부하율과 시도 호 실패 수에서 우수한 성능을 보였으며 SCP 과부하 시 SS NO.7 네트워크 트래픽의 감소를 가져왔다.

ABSTRACT

In traditional approaches to IN load control, there has been a focus on the protection of individual SCPs. On the other hand, the load control of network level can be achieved by using a mobile agent technology, which is a software element responsible for moving from one system to another and performing their task. In this thesis, we propose an mobile agent mechanism and two mobile agent migration procedures and analyze the performance of the Call Gapping mechanism and the mobile agent mechanism. The mobile agent mechanism showed that the SCP load control rate and attempt the number of attempted call failure were better than those of Call Gapping mechanism. Also, the amount of SS NO.7 traffics was reduced in the case of an overload condition.

1. 서론

지능망 서비스는 호 루팅과 직접 관련 없는 가상 번호를 가입자 번호로 사용함으로써 지역에 관계없이 전국 공통으로 사용할 수 있고, 하나의 가상번호에 복수의 실제 가입자 번호를 두어 서비스의 사용 지역, 일시, 용도 등에 따라 착신 가입자의 번호변역이 가능하며 서비스의 종류에 따라 과금 형태를

다양하게 함으로서 착, 발신자 모두에게 이익을 줄 수 있는 서비스이다. 지능망 서비스는 여러 가지 고객 요구에 맞는 새로운 서비스를 효율적이고 빠르게 제공하는데 목적이 있다¹⁾.

최근의 지능망 서비스는 그 수의 증가뿐만 아니라 서비스 종류도 다양해져 한정된 지능망 자원으로는 고객 요구를 만족시키기가 어려워졌다. 따라서 지능망 서비스의 품질뿐만 아니라 네트워크의 효율

* 하나로통신(주)(ij0307@hanaro.com)

** 광운대학교

논문번호 : 010358-1127, 접수일자 : 2001년 11월 27일

** 이 논문은 2001년도 광운대학교 교내 학술 연구비 지원에 의해 연구되었음.

적인 사용을 위해 지능망 부하제어 메커니즘의 선택은 중요하다²⁾. 전통적인 지능망 부하제어 메커니즘인 Call Gapping 메커니즘은 SSP에서 제공되는 Call Gapping 기능을 사용하여 특정 서비스, 특정 발신번호, 특정 착신번호 등을 대상으로 서비스를 제안하는 기능을 제공한다. 서비스 관리자에 의해 Call Gapping에 필요한 파라미터를 설정하여 대상 서비스에 대한 SSP로부터의 트래픽을 제한하여 SCP의 부하를 제어한다³⁾⁴⁾. 즉, 개별적인 SCP 보호가 지능망 부하제어의 주요 관점이다. 이러한 제어 방법은 트래픽의 특성이 고려되어 있지 않기 때문에 특정 서비스가 같은 시간에 다량으로 발생되어 SCP에 과부하를 발생시킬 경우 전체 QoS (Quality of Service)를 저하시킬 수 있는 문제점을 가지고 있다.

본 논문은 지능망 전체 네트워크 자원 활용의 최적화를 통한 부하제어를 주요 관점으로 하여 이동 에이전트 기술을 이용한 지능망 과부하 제어 메커니즘과 이동 절차를 제안하였다. 제안한 이동 에이전트 메커니즘은 SCP 과부하 발생 시 SCP기능을 갖는 서비스 로직과 데이터가 SSP로 이동하여 SCP 기능을 대신함으로써 특정 서비스에 의한 전체 QoS 저하 문제를 해소 할 수 있으며, SCP 부하를 SSP로 분산 시켜 지능망 전체 네트워크 자원을 효율적으로 활용할 수 있다. 제안한 메커니즘은 이동 에이전트 기술의 이용으로 가능해지는데 이동 에이전트는 실행환경이 구현된 모든 시스템으로의 이동이 가능하고 독자적으로 작업 수행이 가능한 소프트웨어이다⁵⁾⁶⁾.

본 논문의 구성은 다음과 같다. 제 2장에서는 기존 Call Gapping 메커니즘과 제안한 이동 에이전트 메커니즘 및 이동 에이전트 이동 절차에 대해 설명하였다. 제 3장에서는 지능망 호 설정 절차를 Call Gapping 메커니즘에서의 지능망 호 설정 절차와 이동 에이전트 메커니즘에서의 지능망 호 설정 절차로 나누어 기술하였다. 제 4장에서는 시뮬레이션 결과와 성능을 분석하였다. 마지막으로 제 5장은 결론 부분이다.

II. 지능망 부하제어 메커니즘

2.1 Call Gapping 메커니즘

Call Gapping 기능은 SSP에서 제공되는 Call Gapping 기능을 사용하여, 특정 서비스, 특정 발신번호, 특정 착신번호 등을 대상으로 서비스를 제한

하는 기능을 제공한다. 서비스 관리자는 Call Gapping에 필요한 파라미터를 설정하여 대상 서비스에 대한 SSP로부터의 트래픽 인입을 제한함으로써 전체적인 지능망 시스템의 성능을 유지시킬 수 있다. 즉, Call Gapping 메커니즘은 SCP를 보호하는 관점에서 SCP에 과부하가 발생되면 SSP에서 서비스 시도 호를 제한한다. 이 메커니즘은 지능망 서비스의 다양화에 따른 트래픽 특성을 고려하지 않고 있어 만약 여러 SSP에서 동시에 특정 서비스 호가 발생되어 SCP의 과부하를 유발했을 경우 SSP의 시도 호 제한에 의해 다른 서비스호가 제한될 수 있는 문제점을 가지고 있다. 그림 1은 Call Gapping 메커니즘의 흐름도이고, 그 동작과정은 다음과 같다.

- 1) Step 0 : SCP의 부하를 주기적으로 감시
- 2) Step 1 : SCP의 부하가 과부하 상태일 경우 Call Gapping Duration과 gap값이 포함된 Call Gapping 메시지를 SSP로 송신
- 3) Step 2 : Call Gapping 메시지를 수신한 SSP는 Duration동안 조건에 맞는 호에 대한 Gapping을 실행. 조건을 만족하는 첫 호는 SCP로 전달
- 4) Step 3 : Step2을 Call Gapping Duration간 반복 수행

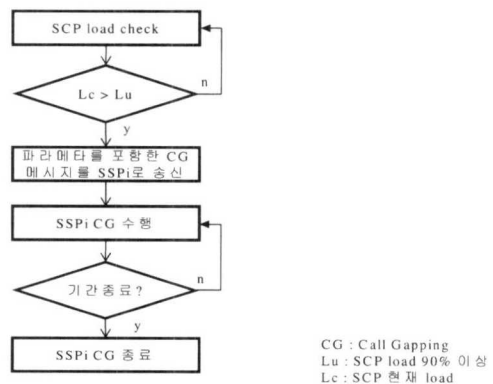


그림 1. Call Gapping 메커니즘

2.2 트래픽 조건에 따른 SCP부하의 변화

지능망 서비스의 종류가 다양해짐에 따라 트래픽의 유형도 다양해지고 있다. 그림 2는 트래픽 조건에 따른 SCP가 처리할 수 있는 호수의 변화를 보여주고 있다. 우리는 그림 2에서 SCP가 처리할 수 있는 서비스를 서비스 A, B 그리고 C로 가정하고,

각 서비스의 처리 시간을 1, 5, 20[msec]로 가정한다. 트래픽 유형 X에서 서비스 A의 서비스 시도율은 80[call/sec], 서비스 B는 10[call/sec], 그리고 서비스 C도 10[call/sec]이라고 하자. 이것은 총 서비스 시도율이 100[call/sec]이다. 이 상황에서, SCP 시도 부하는 0.33이고 처리할 수 있는 호의 수 100[call/sec]이다. 다음으로 트래픽 유형 Y에서 서비스 A의 서비스 처리율은 10[call/sec]이고, 서비스 B도 10[call/sec], 그리고 서비스 C는 80[call/sec]이라고 하자. 이것의 총 시도율은 100[call/sec]이다. 그러나 SCP 시도 부하는 1보다 큰 1.66이다. 그래서 SCP는 100[call/sec]를 처리할 수 없다. 따라서 트래픽의 유형에 따라 SCP가 처리할 수 있는 호의 수는 예측하기 어렵다. 트래픽 유형 Y에서 발생된 SCP 과부하는 66%이고, 트래픽 유형 X에서 SCP는 과부하가 아니다. 만약 Call Gapping이 SCP 부하율 90%에서 이루어 진다면 트래픽 유형 Y에서는 SSP 서비스 호의 76%가 Gapping된다. 즉, 서비스 C로 인해 서비스 A, B의 서비스 시도 호가 Gapping 된다. 이렇게 Call Gapping 메커니즘은 트래픽 유형 Y에서와 같이 특정 서비스로 인한 과부하 발생으로 전체 QoS가 저하될 수 있다^[7].

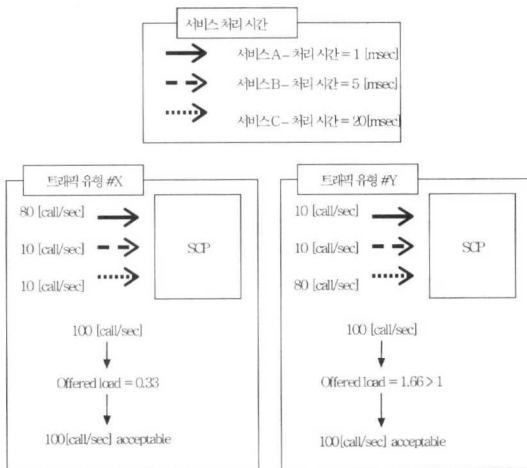


그림 2. 예측 불가능한 호의 수

2.3 제한한 이동 에이전트 메커니즘

이동 에이전트를 이용한 과부하 제어 메커니즘은 SCP 과부하 발생 시 SCP부하에 가장 큰 영향을 주는 SSP를 찾아 SCP 기능을 이동 시켜 SSP로 이동된 만큼의 SCP부하를 감소시킨다. 또한 Call Gapping 메커니즘은 개별적인 SCP 관점에서 지능

망 부하 제어를 했는데 제한한 이동 에이전트를 이용한 과부하 제어 메커니즘은 SCP 부하를 SSP로 분산시킴으로써 지능망 네트워크 관점에서 지능망 부하 제어를 할 수 있다.

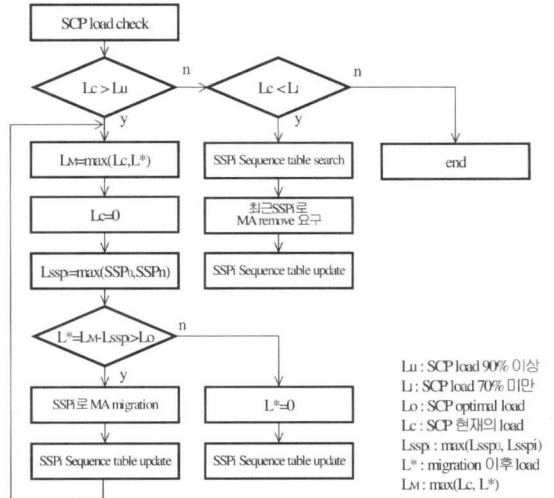


그림 3. 이동에이전트 부하제어 메커니즘

그림 3은 이동 에이전트를 이용한 과부하 제어 메커니즘의 흐름도이고, 그 동작과정은 다음과 같다.

- 1) Step 0 : SCP 부하 주기적 감시
- 2) Step 1 : SCP 부하가 Overload일 경우 이동 에이전트 이동절차를 실행하고 SCP 부하가 Underload일 경우 이동 에이전트 삭제 절차 실행
- 3) Step 2 : 이동 에이전트를 이동시킬 경우 SCP Queue의 대기호 중 비율이 가장 높은 SSP를 식별하여 이동 에이전트를 이동 후 Sequence table update 실행
- 4) Step 3 : 이동 에이전트 삭제할 경우 Sequence table에서 가장 최근 SSP로 remove 메시지 송신 후 Sequence table update 실행

이동 에이전트 메커니즘은 크게 이동 에이전트 이동 절차와 삭제 절차로 이루어진다. SCP 부하를 주기적으로 감시하여 L_c 값이 L_u 값 보다 클 경우, 에이전트 이동절차를 수행하고 L_c 값이 L_l 값 보다 작은 경우에는 에이전트 삭제 절차를 수행한다. 이동 절차는 L_c 와 L^* (초기값=0)를 비교해 최대값을 L_m 에 할당한다. 그 다음은 L_c 를 초기화하고 L_{ssp} 값에 SCP Queue 대기 호 중 비율이 가장 높은 SSP를 찾는

다. LM에서 L_{sspi} 를 뺀 값이 L_0 보다 작아질 때까지 SSP로의 에이전트 이동 및 SSPi Sequence table을 수정, 에이전트 이동 절차가 반복 수행된다. LM에서 L_{sspi} 를 뺀 값이 L_0 보다 작으면 L^* 를 초기화하고 SSPi Sequence table을 수정한다. 에이전트 삭제 절차는 L_c 값이 L_i 값 보다 작은 경우 SSPi Sequence table을 찾아 가장 최근에 이동된 에이전트를 삭제하기 위해 SSP로 삭제 요구를 한다. 삭제 후 SSPi Sequence table을 수정한다.

2.4 제안한 이동 에이전트 이동절차

이동 에이전트의 이동 절차는 자동 및 수동 절차를 제안하였다. 그림 4의 자동 절차는 지능망 과부하의 예측이 불가능 할 경우, 즉 불특정 시간이나 불특정 SSP에서 발생한 서비스 호가 SCP 과부하를 유발했을 경우 SCP 과부하 레벨에 따라 이동 절차가 수행된다. SCP는 SSP_{target}와 SSP_{other}로부터 지능망 Service request 메시지를 수신한다. SSP_{target}으로부터 수신한 Service request 메시지에 의해 SCP의 부하가 90%이상이면 SCP는 SCP의 부하가 90% 미만인 때까지 Queue의 대기 호 중 높은 비율의 대기 호를 갖는 SSP_{target}순서로 이동 에이전트를 이동시킨다. 이동 에이전트를 수신한 SSP_{target}은 Service request 메시지를 SSP_{target}로 이동한 에이전트로 보낸다. 즉 SCP의 부하가 SSP_{target}의 이동 에이전트로 이동되며 SCP에 줄어든 Queue에는 SSP_{other}로부터 수신되는 Service request 메시지들이 대기하게 된다. SCP 부하가 69% 미만으로 내려갈

경우 SCP는 이동 에이전트 삭제 메시지를 SSP_{target}으로 보내게 되는데, 이때 이동 에이전트가 삭제 될 SSP_{target}은 마지막으로 이동 에이전트가 도착한 SSP_{target}이다. 마지막으로 이동한 에이전트를 삭제하는 이유는 가장 마지막에 이동한 에이전트의 SSP_{target}가 SCP의 Queue를 가장 적게 점유한 SSP이므로 이동 에이전트 삭제 후에 SSP로부터 수신될 수 있는 Service request 메시지를 최소화하여 SCP 부하의 급격한 증가를 막기 위해서이다.

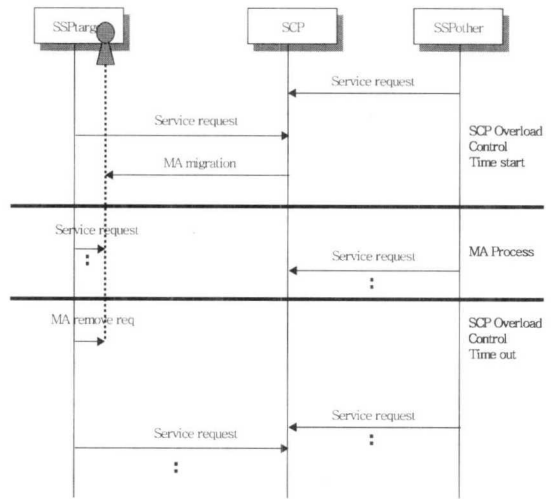


그림 5. 이동 에이전트 수동 이동절차

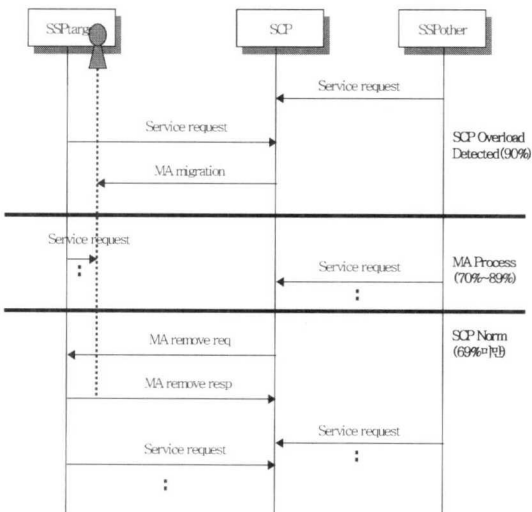


그림 4. 이동 에이전트 자동 이동절차

그림 5의 수동 절차는 지능망 서비스 트래픽의 유형이 예측 가능한 경우, SCP 운용자가 이동 에이전트 이동에 필요한 SSP_{target}과 이동시간을 등록한다. SCP는 부하제어 타이머가 시작되면 이동 에이전트를 SSP_{target}으로 이동시킨다. 이동 에이전트를 수신한 SSP_{target}은 Service request 메시지를 에이전트로 보낸다. SCP 부하제어 타이머가 종료되면 에이전트는 SSP_{target}는 이동 에이전트를 삭제시킨다.

III. 지능망 호 설정 절차

3.1 일반적인 지능망 호 설정 절차

그림 6은 이동 발신 가입자가 이동 착신 가입자로 지능망 호 설정이 이루어지는 일반적인 절차이다. 발신 측 SSP_i에서 SCP로 호 설정을 위해 IN service Query Req 메시지를 보낸다. SCP는 착신 가입자에 대한 정보를 확인한 후 IN service Query

Resp 메시지를 SSP_o로 송신한다. 착신 가입자에 대한 실제 착신 번호를 받은 발신 측 SSP_o는 VLR_o로 착신 가입자의 위치 정보를 위해 Location Query Req 메시지를 보낸다. VLR_o는 다시 HLR로 Location Query Req 메시지를 보내고, HLR로부터 수신한 Location Query Resp 메시지를 발신 SSP_o로 송신한다. 발신 SSP_o는 착신 가입자가 위치해 있는 착신 SSP_i로 Call Setup Req 메시지를 송신한다. SSP_i는 VLR_i와 Routing information 정보를 주고받은 후 SSP_o로 Call Setup Resp 메시지를 송신하게 된다.

그림 7은 유선 가입자에서 유선 착신 가입자로 호 설정이 이루어지는 절차이다. 발신 측 SSP_o는 SCP로 IN service Query Req를 송신한다. SCP는 착신 가입자에 대한 정보를 확인한 후 SSP_o로 IN service Query Resp 메시지를 송신한다. SSP_o는 실제 착신번호가 있는 SSP_i로 Call Setup Req 메시지를 송신하고 SSP_i는 Call Setup Resp 메시지로 응답한다.

그림 6, 7은 Call Gapping 메커니즘에서의 지능망 호 설정 절차이다. 지능망 호 설정 절차에서 지능망 과부하를 유발하는 메시지는 IN service Query Req 메시지와 IN service Query Resp 메시지이다. Call Gapping 메커니즘인 경우 SCP에 과부하가 발생되면 SCP는 SSP_o로 시도호 Gapping 요구 메시지를 송신한다. SSP_o에서 실행되는 시도호의 Gapping은 트래픽 유형의 변화를 예측하기가 어렵다. 따라서 특정 서비스에 의해 다른 서비스들의 QoS가 낮아질 수 있다.

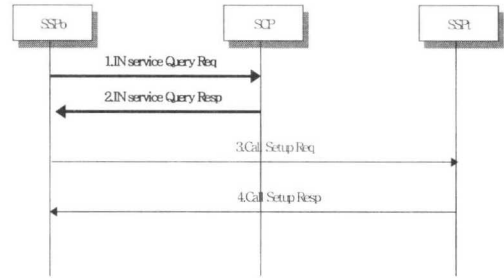


그림 7. 유선 네트워크에서 지능망 서비스 호 설정 절차

3.2 에이전트 이동 후 지능망 호 설정 절차

그림 8은 이동 에이전트 메커니즘을 적용해 SCP에 과부하가 발생되어 이동 에이전트가 SSP_o로 이동된 이후 무선 가입자가 유선 착신지능망 서비스를 받기 위한 호 설정 절차이다. SSP_o는 착신 가입자 정보를 SSP_i에 있는 에이전트로 IN service Query Req 메시지를 보내 확인한다. 착신 정보를 확인 후 VLR_o로 Location Query Req 메시지를 송신하고, HLR를 통해 Location Query Resp 메시지를 수신한다. 착신 SSP_i를 확인 한 SSP_o는 Call Setup Req 메시지를 SSP_i로 송신하고, Call Setup Resp 메시지를 수신한다. 그림 8은 앞의 그림 6, 7번의 IN service Query Req 메시지와 IN service Query Resp 메시지를 SCP로 송신하지 않고 SSP_o내의 이동 에이전트를 통해 착신 가입자 정보를 얻는다. 지능망 호 설정 시 SCP 과부하를 유발하는 두 메시지를 없애게 되면 SCP 부하의 감소뿐만 아니라 SS NO.7 네트워크 트래픽의 감소가 가능하다.

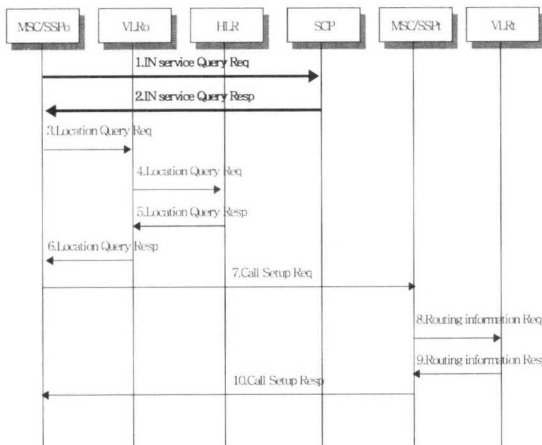


그림 6. 무선 네트워크에서 지능망 서비스 호 설정 절차

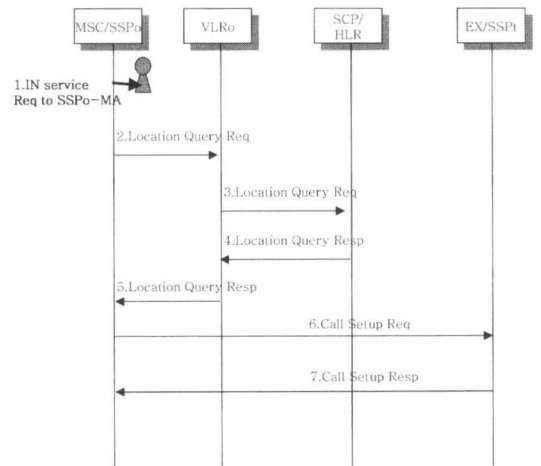


그림 8. 유/무선 통합 네트워크에서 지능망 서비스 호 설정 절차

IV. 시뮬레이션 결과 및 성능분석

본 논문에서 Call Gapping 메커니즘과 이동 에이전트 메커니즘의 성능 분석은 표 1의 파라메타와 값을 이용해 수행하였다. 시뮬레이션을 위한 지능망 네트워크의 컴퍼넌트들은 SS NO.7네트워크 기반에서 SCP와 SSP로 가정하였으며, 각 컴퍼넌트들에는 이동 에이전트가 생성, 실행, 삭제될 수 있는 에이전트 실행환경이 구현되어 있다. SCP 개수는 1개로 하였고, SCP의 Queue 크기는 10000으로 하였으며, SCP 부하의 최대값을 0.81이상으로 하였고 최소값을 0.69이하로 하였다. SCP 부하의 최대값은 이동 에이전트 메커니즘에서 이동 에이전트가 SSP로 이동하는데 필요한 파라메타이고 SCP 부하의 최소값은 이동에이전트를 삭제하는데 필요한 파라메타다. SCP의 적정 부하는 0.80으로 가정하였다. SSP의 파라메타와 값에서 SSP는 20개로 하였고, SSP Queue 크기는 500으로 하였다. SSP로 수신되는 시도 호는 일반 호와 지능망 서비스 호로 나눌 수 있는데 일반호의 Queue와 지능망 서비스 호 Queue는 분리되어 있어 일반 호에 의한 지능망 서비스호의 제한은 없는 것으로 가정하였다. SSP 서비스 호의 합은 20개의 SSP에서 SCP로 요구하는 서비스 호의 총 개수를 의미하며 8000 ~ 10000호까지 100호 단위로 Level 1 ~ Level 21까지 나누었다.

Queueing Model의 주요 요소는 SS NO.7네트워크 내의 SCP노드와 SSP 노드들이다. 표 1에서와 같이 SCP의 Queue는 길이가 10000이고 FIFO방식으로 동작한다. SSP의 Queue는 길이가 500이고 SCP와 같이 FIFO방식으로 동작한다.

표 1. 시뮬레이션 파라메타

SCP		SSP	
파라메타	값	파라메타	값
SCP 개수	1개	SSP 개수	20개
SCP Queue Size	10000	SSP Queue Size	500
SCP 부하	최대값: 0.81 이상 최소값: 0.69 이하	SSP 서비스 호의 합	Level1: 800 Level11: 900 Level2: 800 Level12: 900 Level3: 800 Level13: 900 Level4: 800 Level14: 900 Level5: 800 Level15: 900 Level6: 800 Level16: 900 Level7: 800 Level17: 900 Level8: 800 Level18: 900 Level9: 800 Level19: 900 Level10: 800 Level20: 900 Level21: 1000
SCP 적정 부하	0.80		

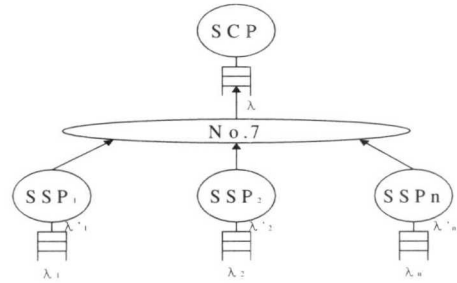


그림 9. 지능망 queueing model

$\lambda_1 \sim \lambda_n$ 은 초당 사용자로부터 SSP로 들어오는 서비스 시도 호를 의미하고, $\lambda'_1 \sim \lambda'_n$ 는 Call Gapping 메커니즘의 경우는 Gapping 이후 SCP로 요구하는 서비스 호를 의미하며, 이동 에이전트 메커니즘의 경우는 SSP에서 이동 에이전트가 처리하는 서비스 호 외에 SCP로 요구하는 서비스 호를 의미한다. λ_{tot} 은 λ'_1 에서 λ'_n 까지의 총 합을 의미하며 8000 ~ 10000호를 100호 단위로 나누었다. 이들은 포아송 분포를 가지며 호의 우선 순위는 없는 것으로 가정하였다.

그림 10은 지능망 과부하 발생 시 Call Gapping 메커니즘과 이동 에이전트 메커니즘이 변화되는 SSP 서비스 요구 호 합에 따라 SCP의 부하를 유지하는 성능을 비교해 보여준다. 그림 10에서 Call Gapping 메커니즘의 경우 SSP 서비스 요구 호 총 수가 SSP서비스호의 합이 Level1과 21인 경우를 제외하고는 적정 부하 80%를 유지하지 못하고 있다. 이 경우 Call Gapping 메커니즘은 SCP 적정 부하 80%를 넘는 호들은 모두 실패된다. 반면에 이동 에이전트 메커니즘의 경우 SSP서비스호의 합 Level 전체에서 적정 부하 80%를 넘지 않는 수준을 유지하고 있다. 따라서 기존 Call Gapping 메커니즘에서 지능망 과부하로 발생될 수 있는 다른 서비스의 QoS 문제를 해소 할 수 있다.

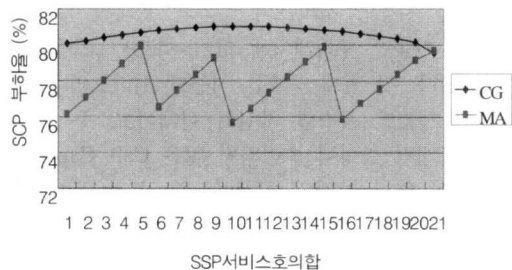


그림 10. SCP 부하율 비교

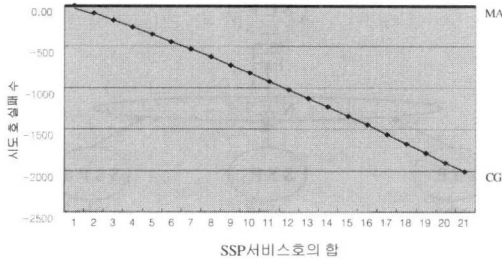


그림 11은 SCP가 적정 부하 80%를 유지한다고 할 때 SCP로 인입 되는 서비스 요구 호의 실패 수를 비교해 보여준다. 즉, 그림 11의 x축인 SSP서비스호의 합 Level이 높아짐에 따른 Call Gapping 메커니즘과 이동 에이전트 메커니즘의 시도호 실패수를 비교하였다. Call Gapping 메커니즘은 SCP의 적정 부하 80%를 넘을 경우 SSP에서 시도호가 Gapping됨에 따라 적정 부하 80%넘는 호들은 모두 실패되고, 이동 에이전트 메커니즘은 적정 부하율 80%를 넘을 경우 SSP로 SCP기능이 이동되므로 SSP Queue에 대기하는 서비스 호는 SSP에서 처리하게 되어 SSP Queue내의 대기 호는 모두 처리가 가능하게 된다. 또한 이동 에이전트 메커니즘을 적용할 경우 적정 부하율 80%를 넘은 호는 SSP에서 처리되므로 SSP에서 처리되는 호 수 만큼은 SSP에서 SCP로의 서비스 요구 호 수가 줄어들게 되어 SSP와 SCP사이의 SS NO.7 네트워크의 트래픽이 감소하게 된다.

V. 결론

본 논문에서는 네트워크 차원의 지능망 과부하 제어 메커니즘인 이동 에이전트 메커니즘을 제안하였다. 기존 Call Gapping 메커니즘은 지능망 과부하를 SSP로부터 SCP로 인입되는 트래픽을 제한하여 전체 지능망 시스템의 성능을 유지한다. 고객 요구에 의한 지능망 서비스의 증가와 트래픽 특성의 다양화는 지능망 부하제어에 있어 새로운 문제점을 야기 시켰다. 첫 번째 문제는 지능망 서비스의 증가에 따른 지능망 시스템 자원의 한계이고, 두 번째 문제는 트래픽 특성이 고려되지 않은 Call Gapping 메커니즘의 QoS 보장의 한계다. 즉, 특정 시간 또는 특정 SSP에서 발생하는 서비스 호로 인한 SCP 과부하가 발생할 경우, 기존 Call Gapping 메커니즘은 다른 서비스의 QoS를 보장하기 어렵다.

반면 본 논문에서 제안한 이동 에이전트 메커니

즘은 SCP 과부하 발생 시 SCP 기능을 가진 서비스 로직과 데이터가 SSP로 이동하여 실행됨으로써 지능망 자원의 효율성을 높일 수 있다. 또한 제한한 이동 에이전트 이동 절차는 자동절차와 수동절차에서 트래픽 특성을 고려하였다. 시뮬레이션을 통해 지능망에서 SCP 부하율과 서비스 요구 호 실패 수로 Call Gapping 메커니즘과 이동 에이전트 메커니즘의 성능을 분석한 결과 SCP 부하율에서 Call Gapping 메커니즘의 경우 SSP서비스호의 합Level이 변화됨에 따라 적정 부하율을 유지하지 못하였고, 이동 에이전트 메커니즘은 SSP 서비스호의 합Level 변화에 따라 적정 부하율 이하에서 SCP부하율을 유지하였다. 또한 서비스 요구 호 실패 수의 경우 Call Gapping 메커니즘은 SSP 서비스호의 합Level이 변화함에 따라 시도 호의 Gapping에 의해 호 실패 수가 증가하는 반면, 이동 에이전트 메커니즘은 서비스 요구 호 전체를 처리하였으며, 이동 SSP에서 처리된 서비스 요구 호 만큼 SS NO.7 네트워크 트래픽이 감소하는 결과를 가져왔다.

결론적으로 이동 에이전트 메커니즘은 기존 Call Gapping 메커니즘에 비해 지능망 과부하 시 QoS 저하를 최소화 할 수 있으며, 지능망 서비스 증가에 따른 SCP자원의 한계를 SSP 자원의 활용으로 해소할 수 있다.

참고 문헌

- [1] B. Jabbari, "Intelligent Network Concepts in Mobile Communications", *IEEE Communications Magazine*, Vol. 30, pp. 64-69, February 1992.
- [2] B. Carlsson, P. Davidsson and S. Johansson, "Using Mobile Agents for IN Load Control", *Intelligent Network Workshop 2000 IEEE*, pp. 161-169 2000.
- [3] 이용, 송주석, "지능망 구조에서 공정성 우선순위를 보장하는 과부하제어 연구", *한국통신학회 논문지*, pp. 3098-3108, December 1996.
- [4] N. Tsolas, G. Abdo and R. Bottheim, "Performance and Overload Considerations When Introducing IN into an Existing Network", *International Zurich Seminar*, 1992.
- [5] L. Hagen, M. Breugst and T. Magedanz, "Impacts of Mobile Agent Technology on

