유전 알고리즘을 활용한 서비스 개선비용을 최적화하기 위한 QoE Framework에 대한 연구

A Study about QoE Framework for optimizing service improvement cost using genetic algorithm

현 태 환*, 김 화 종**, 오 치 문*, 이 경 현***
Tae-Hwan Hyun, Hwa-Jong Kim, Chi-Moon Oh, Kyoung-Hyun Lee

Abstract

네트워크 서비스에 있어서 품질을 정확히 측정하고 품질지수를 도출하는 것은 매우 중요하다. 종래의 네트워크 서비스에 대한 품질지수는 QoS(Quality of Service)로서 많은 연구가 있어왔다. 하지만 QoS를 통한 사용자 품질지수는 실제 사용자의 서비스 만족도를 효과적으로 표현해 주지 못하였다.

그래서 본 연구에서는 QoE(Quality of Experience)라는 개념을 도입하여, 사용자가 실제로 느낀 서비스의 만족도를 도출하고자 하였다. 사용자의 서비스 만족도를 잘 표현해주는 QoE를 도출하기 위하여 본 연구에서는 QoE Framework를 제안한다. QoE Framework는 framework를 위해 동작하는 agent들로 구성된다. QoE Framework의 agent는 크게 서비스 제공자측과 서비스 사용자 측으로 나눌 수 있는데, 서비스 사용자 측의 agent는 서비스를 이용하다가 사용자가 서비스 불만족을 감지하였을 때, 서비스 제공자측 agent로 불만족을 feedback하는 기능과 불만족이 발생한 시점의 네트워크 상황을 측정하는 기능으로 이루어져 있다. 또 서비스 제공자측 agent는 사용자측 agent로부터 받은 서비스 불만족에 대한 feedback으로부터 QoE도출 알고리즘을 사용하여 최적의 QoE를 도출하는 기능을 한다. 서비스 제공자측 agent의 QoE도출 알고리즘으로 GA(Genetic Algorithm)을 사용하여 최적의 값을 구하게 된다.

QoE framework의 알고리즘에서는 불만족 feedback을 이루는 QoS 파라미터들간의 중요도와 관계를 도출하여, 주어진 서비스 개선 비용으로 최적의 QoE를 도출함으로써, 네트워크 서비스에 대한 사용자의 만족도를 최대로 높이는 결과를 도출하게 된다.

Keywords : QoE, QoS, Gentic algorithm.

I. 서 론

최근 이슈인 IPTV는 초고속 광대역 네트워크를 이용하여 디지털 영상 서비스를 비롯한 다양한 서비스를 제공하는 신개념 서비스이다. 방송과 통신이 서로 융합하면서 방송과 인터넷과 전화를 하나로 아우른 TPS 서비스 구현이나 인터넷포털과 통신사업자의 제휴로 TV포털 사업과 같은 형태의 서비스가 등장하고 있다[1-2].

현재의 초고속 인터넷 서비스는 서비스의 품질을 평가하는 지표로서 오로지 평균 대역폭만을 사용하고 있다. 하지만 IPTV는 서비스의 품질을 보장하기 위한 평가지표로서 평균 대역폭이외에도 많은 파라미터들을 정의하고 있다[3-5]. 그렇기 때문에 IPTV서비스를 현재의 초고속 인터넷망으로 서비스하는 경우 사용자가 체감하는 서비스의 품질을 제대로 반영하지 못하게 된다. 따라서 초고속망으로 IPTV를 시청하는 경우 사용자가 느끼는 서비스의 품질을 향상시키기 위하여 시스템을 이루는 구성요소 중 어떤 것을 얼마나 개선해야 서비스 품질이 최대로 개선될 것인가에 대한 고민을 하지 않을 수 없다. 따라서 이를 위해 QoE를 도입하여 문제를 해결하고자 한다.

하지만 종래의 QoE에 대한 연구는 사용자 만족도 자체를 측정하는데 그치는 면이 있다. [6]의 연구에서는 GAP model이라는 모델을 제시하였고 사용자 만족도를 Good, Acceptable, Poor의 세 등급으로 나누었다. 그리고 각 세 등급의 임계치는 고정적인 상수값이다.

QoS 파라미터인 Bandwidth, Delay, Loss, Jitter에 대한 세 등급의 임계치를 설정하여 최종 QoE를 네 파라미터가 가지는 등급값의 조합으로 나타내었다. 이 연구는 각 등급의 임계치를 고정적인 상수값을 사용하였다는 점에서 지속적으로 변화하는 사용자의 행태를 유연하게 받아 들이기 힘들다는 단점이 있다. 또 [7]의 연구에서는 Bandwidth에 따른 VQM(video quality metric) 점수, 화면 동기화, 채널변경 시간등에 대하여 기술하고 있는데 이들 역시 서비스의 개선에 필요한 서비스 품질지표를 도출하였다고 보기 힘들다.

이렇듯 서비스의 품질 향상을 위하여 품질에 영향을 미치는 파라미터를 정확히 찾아내고 또 그 파라미터의 조합을 통한 QoE의 분석시스템이 사용자의 변화하는 요구에 맞춰줄 수 있는 것이어야 한다는 논지로부터 실시간성있는

*현태환,오치문 : 강원대학교 컴퓨터정보통신공학과
**김화종: 강원대학교 컴퓨터정보통신공학과 정교수
***이경현 : 강원대학교 전기공학과

QoE 분석시스템을 기술하고자 한다.

1장의 서론에 이어, 2장의 In-service simple feedback, 3장의 QoE분석 시스템의 구조, 4장의 Genetic Algorithm, 5장의 제안하는 QoE 분석 시스템에 이어 6장에서 결론을 맺는다.

II. In-service simple feedback

서비스의 품질을 평가하기 위해서 널리 사용되는 것 중 하나는 MOS이다. MOS는 만족도를 흔히 5점으로 표현하는 표준화된 수치화 방법이다. 하지만 MOS를 이용하여 실시간적인 서비스의 품질을 정확히 표현하기에는 무리가 따른다. 일반 사용자가 전체 서비스 중 어느 때 어떤 원인에 의하여 불만족이 있었는지를 모두 기억하여 정확한 답변을 주기란 매우 어려운 일이기 때문이다. 그렇기 때문에 사용자가 서비스를 제공받으면서 서비스에 대하여 불만족을 느끼는 순간의 서비스 상황에 따른 불만족 여부를 보고하는 것이 정확한 데이터에 의한 정확한 결과가 도출되게 하는 방법일 것이다.

그리고 사용자가 서비스 컨텐츠 자체를 즐기는 것에 큰 영향을 주지 않는 범위내에서 서비스의 불만족을 보고하게 하는 방법을 선택하는 것이 좋을 것이다. 사용자는 서비스를 받고자 하는 것이 목적이지 불만족을 보고하는 것이 목적이 아니기 때문이다.

그래서 QoE 분석 시스템은 서비스 도중에 간단한 방식으로 서비스 불만족에 대한 보고를 할 수 있는 형태의 in-service simple feedback을 지원하는 시스템이어야 하겠다.

III. QoE Framework의 구조

스템은 2개의 agent로 구성된다. 하나는 server agent이고, 다른 하나는 user agent이다. 먼저 server agent는 user agent(사용자)로부터 서비스 불만족시 불만족에 대한 보고를 수집하여 GA를 통하여 현재 서비스가 개선되기 위하여 고려할 최적의 결과를 도출하는 역할을 한다. 반면에 user agent는 서비스를 제공받다가 서비스의 품질이 불만족스러울 때 간단한 방식으로 server agent에 불만족 시점을 보고하는 역할을 한다.

IV. Genetic Algorithm

1970년대 초 John Holland에 의해 본격적으로 연구되기 시작한 Genetic Algorithm(이하 GA)는 자연 생태계의 진화과정에서 관찰된 적자생존의 원리를 컴퓨터 알고리즘에 결합하여 정립된 최적화 알고리즘이다. GA는 해결하고자 하는 문제를 유전자로 표현하는 것과 유전자로 표현되는 문제에 있어서 이를 평가하는 적합도함수, 그에 따라 훌륭한 유전자를 후대로 전달하기 위한 선택함수, 그리고 우등한 유전자를 만들어내기 위한 교배, 돌연변이 함수등으로 이루어져 있다.

사용자로부터 실시간성이 있는 데이터를 잘 모았다면 그 다음으로는 "모인 데이터들로부터 서비스에 영향을 미치는 요소들중 어떤 것이 얼마나 영향을 미치는 지"와 같은 의미있는 결과를 도출하는 것이 필요하다.

의미있는 결과를 도출하는 것은 서비스에 영향을 미치는 여러 요소-QoS 파라미터와 같은-들이 어떤 방식으로 서로에게 영향을 주는지, 각 파라미터간의 상호관계를 규명하는 일과 같다.

제한적으로 QoS 파라미터만을 두고 볼 때 Bandwidth, Jitter, Packer Loss의 세 파라미터는 각각 단위가 모두 다르고, 중점적으로 고려해야하는 파라미터의 범위도 각각 다르다. 이렇게 각 파라미터의 단위와 범위가 각기 다른 경우, 이런 파라미터들의 조합을 모두 계산하여 최적의 해를 구한다는 것은 거의 불가능에 가깝다. 그렇기 때문에 확률적으로 추정하는 GA를 통하여 이 문제를 풀어보려고 한다.

1. 알고리즘

일반적으로 어떤 문제에 대한 가능해를 GA에서는 염색체라고 표현하고, 염색체를 이루는 유전정보의 표현형을 유전자라고 한다. GA는 어떤 문제에서 도출될 수 있는 가능해(염색체)를 다수 생성하고 이를 popluation(해집단)이라고 명명한다[8].

1) 초기 집단 생성

초기 해집단을 생성한다. 알고리즘에서 사용할 population의 수를 정해 준다.

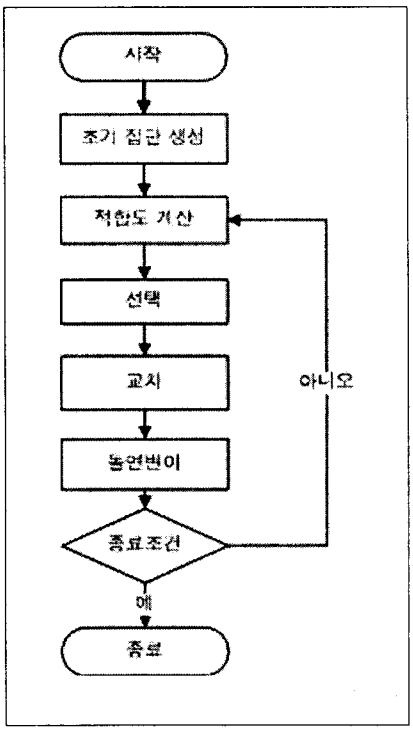


그림1. GA 수행절차
Fig1. Flow chart of GA

2) 적합도 함수

염색체 개체의 생존 능력을 평가한다. 각 개체의 생존능력은 적합도 함수의 반환값에 따라 결정된다.

3) 선택 함수

다음 세대에 유전될 개체를 선택한다. 선택함수는 각 개체가 적합도에 비례하여 선택되는 적합도비례선택(룰렛휠),

해집단내의 해들의 순위를 매긴 후 가장 좋은 해부터 일차 함수적으로 적합도를 배정하는 순위기반 선택 등 다양한 방법이 존재한다.

4) 교배 연산

선택함수에 의해 선택된 두 개체의 유전자 일부를 서로 섞음으로써 새로운 염색체를 생성하는 연산이다. 교배연산에는 1점교배, 2점교배, 균등교배 등 다양한 방법이 쓰인다.

5) 돌연변이 연산

돌연변이 연산은 부모해에 없는 속성을 도입하여 탐색 공간을 넓히려는 목적을 가진 연산으로, 선택된 한 개체에 대하여 유전자 일부를 다른 유전자로 변형시키는 연산이다.

6) 종료 조건

GA의 종료조건은 미리 설정된 세대수를 초과하거나 세대수가 거듭하면서 진화되는 비율이 일정수치 이하일 때 종료할 수 있다.

V. 제안하는 QoE Framework

사용자의 만족도를 향상시키기 위해서는 서비스 품질에 영향을 미치는 파라미터들을 정확히 찾아내고 파라미터간의 관계를 규명하는 것이 필요하다. server agent는 사용자로부터 서비스 도중에 간단한 불만족 보고를 받아 GA를 활용하여 결과를 도출하게 된다. 이때 사용자로부터 보고된 데이터들은 DB에 기록된다.

본 연구에서는 GA의 염색체를 트리기반 염색체를 사용한다. 이는 서비스 품질에 영향을 미치는 파라미터는 다양하고 이들은 복잡한 관계로 얽혀있다는 가정으로부터 출발한 것이다.

1. 유전자형

하나의 해(염색체)는 트리로 이루어 진다. 하나의 노드는 하나의 파라미터와 대치된다. 예를 들면 그림2에서 P는 bandwidth와 A는 delay와 대치될 수 있다. 표 1은 노드를 구성하는 요소를 나타낸다.

표1에서 최소값과 최대값은 사용자가 서비스 도중 서비스의 품질에 대하여 불만족을 느꼈을 때 불만족을 보고하게 되는데 이때 보고된 데이터들중 최소값과 최대값을 의미한다. Threshold는 초기 해집단 생성시 랜덤하게 설정되지만, 사용자의 불만족 보고 횟수가 증가함에 따라 점점 변화한다. 예를 들면 최초 A의 Threshold가 100ms였는데, 사용자의 불만족 보고 데이터의 A(delay)값이 40ms인 데이터가 대단히 많이 수신된다면 최초 100ms에서 점차 내려가는 양상을 보이는 것이다. 유사하게 각 노드의 weight도 랜덤하게 생성된다.

즉, 노드는 어떤 파라미터와 대응되며 해당 파라미터를 수치화하여 최소, 최대값등을 설정할 수 있으며 서비스가 불만족할때마다 보고받은 값들로 만족과 불만족을 구분짓는 threshold를 생성할 수 있는 모든 경우에 대해서 적용이 가능하다. 그리고 하위노드는 상위노드에 영향을 받는다.

표 1. 노드를 구성하는 요소
Table1. Elements which composes a node

구성 요소	설 명
Minimum	파라미터가 가지는 최소값. 서비스 불만족에 따라 보고된 데이터들의 최소값
Maximun	파라미터가 가지는 최대값. 서비스 불만족에 따라 보고된 데이터들의 최대값
Threshold	만족/불만족을 구분짓는 임계치. 사용자에 의하여 불만족 시점에 대한 보고가 많아짐에 따라 점차 최적치에 수렴한다.
weight	알고리즘에서 생성한 파라미터의 가중치. 해당 파라미터의 단위를 가지면서 최소값과 최대값 사이에서 설정되는 **일반 가중치**와 0과 1사이 중 하나의 실수값으로 설정되는 **정규화된 가중치**로 구성된다.

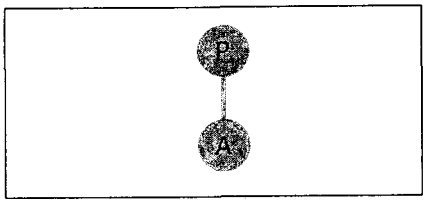


그림2. 염색체 예시
Fig2. Sample of Chromosome

그림2는 노드 A가 노드 P의 하위노드인 트리이다. 위 그림2는 다음의 코드과 같이 쓸 수 있다.

```
if( f(P)==true ){
  if( f(A) == true){
    QoE=true;
  }
}
```

2. Threshold의 갱신

노드의 Threshold는 최초 랜덤하게 설정되고, 사용자의 불만족 보고가 많아짐에 따라 점차적으로 불만족 최적값에 수렴하는 형태로 변경되어 간다.

$$T=\{S_A,S_B,....S_n\} \quad (1)$$

(1) 집합 T는 S_A, S_B 등을 원소로 가지는 전체 집합(Database)이다.

$$S_A=\{A_0,A_1,...,A_n\},S_B=\{B_0,B_1,...,B_n\} \quad (2)$$

(2) 집합 S_A는 DB내의 파라미터 A의 모든 값을 원소로 가지는 집합이다.

$$Sum=\sum_{i=0}^{n} f(A_n) \quad (3)$$

(3) Sum은 파라미터 A_i에 대하여 A의 임계치인 Th_A를 만족하는 개수이다.

$$f(A_i)=\begin{cases}1,g(A_i,Th_A)\\0,else\end{cases} \quad (4)$$

(4) $f(A_i)$는 A_i가 Th_A를 만족하면 1, 아니면 0을 반

환하는 함수이다.

$$g(A,Th)=\begin{cases}A<Th, A\text{의 최소값이 유리할 때}\\ A>Th, A\text{의 최대값이 유리할 때}\end{cases} \quad (5)$$

(5) $g(A,Th)$는 파라미터 A가 최소값이 최적해에 유리한지, 최대값이 유리한지를 판단하는 함수이다.

$$new_Th_A=\frac{(Sum-old_Th_A)}{Sum} \quad (6)$$

(6)은 (3)에서 구한 Sum값을 바탕으로 해당 파라미터에 대한 새로운 임계치를 구하는 식이다.

3. 적합도 함수

본 연구에서 제안하는 방식이 사용자로부터 보고받은 불만족건을 바탕으로 서비스의 개선에 도움이 되는 결과를 도출하는 것이었는데, GA에는 적합도 함수에 사용자의 데이터가 사용된다.

기본적으로 적합도 함수에서는 각 노드의 weight가 노드의 threshold를 만족하면 해당 노드의 weight만큼 fitness value에 누적한다. 이때 상위노드의 weight가 상위노드의 threshold를 만족하지 못했다면 fitenss value에 해당 노드의 weight를 누적하는 것은 취소된다. 적합도 함수에서는 염색체 트리의 형태에 따라 적합도 함수의 판단식이 달라지게 된다. 트리는 그림2와 같은 기본형, 독립형, 종속형 등 다양한 형태로 존재 할 수 있다.

1) 기본형 염색체 트리

기본형은 그림2와 같이 임의의 노드 A가 노드 P의 하위노드로 위치하는 경우로 정의한다. 노드 P가 먼저 만족된 후 노드 A가 만족되었을때, fitness value에 노드 A의 가중치를 누적한다. 이 때 각 노드를 이루는 파라미터의 범위와 단위가 서로 다르므로 누적하는 값은 노드의 정규화 가중치를 사용한다. 노드의 만족은 5-2절의 식(4)의 $f(A_i)$가 반환하는 값으로 알 수 있다.

염색체의 트리가 기본형일 때 적합도 함수는 다음과 같이 정의한다.

```
double FitnessFunction(P, A, ... ) {
    fit=0.0
    if( f(P) ){
        if( f(A) ){
            fit += A.normalizedWeight;
        }
    }
}
```

위 FitnessFunction 함수안에서 f() 함수는 5-2절의 식(4)의 $f(A_i)$와 같다. 이하 독립형, 종속형의 적합도함수에서 나타나는 f()함수도 동일하다.

2) 독립형 염색체 트리

그림3은 독립형 염색체 트리를 나타낸다.

독립형 염색체는 기본형 염색체의 노드 A와 같은 부모노드를 가지는 다른 노드 B가 존재하는 경우로 정의한다. 독립형은 노드 A와 B가 서로 독립적으로 작용한다. 염색체 트리가 독립형일때 적합도 함수는 다음과 같다.

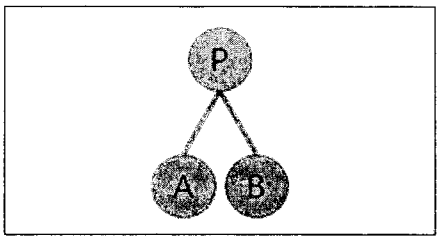


그림3. 독립형 염색체 예시
Fig 3. Sample of Independent Chromosome

```
double FitnessFunction(P, A, B, ... ) {
    fit=0.0
    if( f(P) ){
        if( f(A)){
            fit += A.normalizedWeight;
        }
        if( f(B)){
            fit += B.normalizedWeight;
        }
    }
}
```

3) 종속형 염색체 트리

그림 4는 종속형 염색체 트리를 나타낸다. 종속형은 기본형의 확장이라고 볼 수 있다.

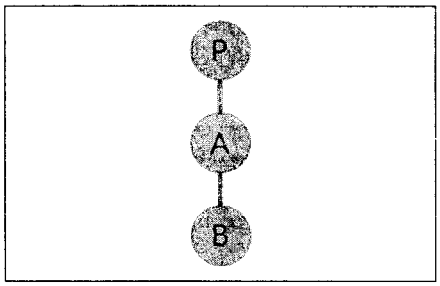


그림4. 종속형 염색체 예시
Fig 4. Sample of Dependent Chromosome

종속형 염색체 트리가 기본형의 확장인 만큼 적합도 함수도 기본형과 유사하다. 다음은 종속형 염색체 트리의 적합도 함수이다.

```
double FitnessFunction(P, A, B, ... )
{
    fit=0.0
    if( f(P) ){
        if( f(A)){
            fit += A.normalizedWeight;
        }
    }
    if( f(A) ){
        if( f(B)){
            fit += B.normalizedWeight;
        }
    }
}
```

4) 교배 / 돌연변이 연산

본 연구의 트리기반 GA에서는 교배연산으로써 depth-dependent crossover(이하 DDC)[9]를 사용하였다. 일반적인 트리에서 crossover를 할 때, 선행되는 selection 함수가 특정 depth(트리의 가장 하위 말단 depth)의 노드에 편중되는 성향이 있는데, DDC는 트리의 각 depth에 일정의 확률값을 주어 전체적으로 고른 확률분포를 보이는 것이 특징이다. 또, 돌연변이 연산은 선택된 노드의 파라미터와 그에 따른 Threshold/Weight를 변경한다.

5) 알고리즘 수행절차

(1) 세대수를 나타낸다. 하나의 해집단(population)이 몇번의 진화를 거듭하는 가를 나타낸다. 위 알고리즘에서는 50회이다.

(2) 해집단의 크기를 나타낸다. 하나의 염색체가 50개가 모여 하나의 해집단을 형성한다.

(3) CGene이라는 데이터형을 가지는 염색체이다. 이 염색체는 트리형태의 자료구조를 내재하고 있다.

(4)~(7) 초기 해집단을 생성하고, 해집단의 최초 적합도를 산출한다.

(8) generation(50회)만큼 반복하면서 진화과정을 거친다.

(9) population(50회)만큼 반복하면서 염색체에 대한 GA연산을 수행한다.

(10) 랜덤함수로 0부터 1사이의 확률값을 구한다.

(11)-(15) 확률값을 비교하여 교배연산을 수행하게 된다. 선택함수를 사용하여 해집단에서 2개체를 선택하고 이를 트리기반 Crossover함수를 통해 교배연산을 수행한다.

(16)-(18) 확률값과 비교하여, 돌연변이 연산을 수행한다.

(20)-(22) 한 세대의 진화가 끝나면 새로운 세대로 구세대를 갱신하여 알고리즘을 계속 수행해 나간다.

```
(1)   generation=50
(2)   population=50
(3)   CGene G[population];

(4)   for(i=0; i<population; i++){
(5)      G[i].Initialize();
(6)      G[i].Fitness();
(7)   }

(8)   for(g=0; g<generation; g++){
(9)      for(i=0; i<population; i++){
(10)        r=Rand(0,1);
(11)        if(r<cRate){
(12)           lhs=Selection(G, population);
(13)           rhs=Selection(G, population);
(14)           newG[i]=Crossover(G[lhs], G[rhs]);
(15)        }
(16)        if(r<mRate){
(17)           newG[i]=Mutation(G[i]);
(18)        }
(19)     }
(20)     for(i=0; i<population; i++){
(21)        G[i]=newG[i];
(22)     }
(23)  }
```

VI. 결 론

본 논문에서는 사용자가 서비스에 대하여 불만족을 감지하였을때의 실시간적인 불만족데이터로부터 현재 서비스에 대하여 어떤 파라미터를 주로 고려해야 하는지 또 파라미터의 상호 관계를 고려해야 하는지 등의 서비스 제공자 입장에서 서비스 개선을 위한 결과를 도출하는 시스템을 제안하였다.

QoE 분석 시스템은 사용자의 만족도를 다룬다는 측면에서 사용자와 떼어질 수 없으며 사용자의 서비스 불만족에 대한 요구는 시시각각 변하며 일정한 수치로 정해질 수 없는 문제이다. 따라서 본 논문에서는 진화하는 알고리즘인 GA로서 QoE 분석문제를 해결하고자 하였다.

QoE에 대한 연구는 사용자가 체감하는 서비스의 품질을 최대로 향상시키는 것에서 출발하지만 결국은 서비스 제공자가 사용자의 만족도를 최대로 끌어올리기 위하여 서비스를 정확히 파악하고 관리하는데 도움을 줄 수 있는 결과를 제공하는 것이 될 것이다.

[참 고 문 헌]

[1] 이상요, "융합시대의 지상파 사업자 전략", IT 산업전망 컨퍼런스, 2007

[2] 김성진, "IPTV 시스템 및 전망", IT 산업전망 컨퍼런스, 2007

[3] ATIS-IIF, ATIS-0800004, ""A Framework for QoS Metrics and Measurements supporting IPTV Services "", October 2006

[4] KT BcN 본부, "IPTV QoS/QoE 측정기술 연구", ICU 학술용역 최종연구보고서, 2007.12

[5] ITU-T FG IPTV-C-0411, "IPTV QoS/QoE Metircs"Prasad Calyam, Eylem Ekici, Chang-Gun Lee, Mark Haffner and Nathan Howes, "A "GAP-Model" based Framework for Online VVoIP QoE Measurement", Journal of Communications and Networks (JCN), 2007

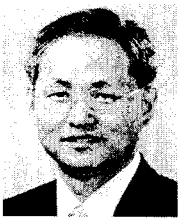
[6] Benny Fallica, Yue Lu, Fernando Kuipers, Rob Kooij and Piet Van Mieghem, "On the Quality of Experience of SopCast", *1st IEEE International Workshop on Future Multimedia Networking FMN'8, Cardiff*, Wales, UK. 2008

[7] 문병로, "유전알고리즘", 두양사, 2003

[8] T. Ito, and S. Sato, *A Self-Tunning Mechanism for Depth-Dependent Crossover, in Advance in Genetic Programming 3*, MIT Press, pp.377-399, 1999

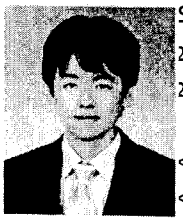
현 태 환

2008년 강원대학교 정보통신공학과 졸업
2008년~현재 강원대학교 컴퓨터정보통신공학과 재학중
<관심분야> QoE, QoS, geneitc algorithm, bio-inspired
<e-mail> j127th@naver.com



김 화 종

1982년 서울대학교 전자공학과 졸업
1984년 KAIST 전기 및 전자과(공학석사)
1988년 KAIST 전기 및 전자과(공학박사)
1988년~현재 강원대학교 IT특성화대학 컴퓨터정보통신전공 교수
<관심분야> QoE, u-city, 최적화
<e-mail> hjkim3@gmail.com



오 치 문

2008년 강원대학교 정보통신공학과 졸업
2008년~현재 강원대학교 컴퓨터정보통신공학과 재학중
<관심분야> QoS/QoE, 차세대 통신망, 알고리즘
<e-mail> 57gate@hanmail.net

이 경 현

1998년 강원대학교 전기공학과 졸업
2000년 강원대학교 전기공학과 졸업(공학석사)
2008년~현재 강원대학교 전기공학과 재학중
<관심분야> QoE, QoS, u-city, framework
<e-mail> meosil@naver.com