

---

# GFR 서비스를 위한 성능 향상된 버퍼관리 알고리즘

## Performance Improved Buffer Management Algorithm for GFR Service

---

조해성  
건양대학교 전자정보공학과  
Hae-Seong Cho(hscho@konyang.ac.kr)

---

### 요약

기존의 버퍼 관리 방법은 TCP 트래픽의 "burstness" 특성으로 인해 손실 없는 전송만으로는 GFR이 요구하는 최소 전송률 보장과 공평성을 제공하기 어렵다.

본 논문에서는 GFR이 요구하는 최소 전송률 보장과 높은 공평성을 제공하기 위하여 기존의 버퍼관리 기법에 퍼지 이론을 적용한 버퍼관리 알고리즘을 제안하였다. 제안된 알고리즘은 태깅정보, 버퍼 점유량, VC의 부하정도로 구성된 세 개의 파라미터를 이용하여 패킷 폐기를 결정한다. 시뮬레이션 결과 MCR의 크기가 커질수록 제안된 알고리즘의 수율과 공평성이 우수하였으며, 스위치에서 버퍼의 크기 변화에서도 Double-EPD나 DFBA는 비슷한 성능을 보였고, 제안된 알고리즘은 좋은 수율과 공평성을 제공한다.

■ 중심어 : | GFR | 퍼지 | 버퍼관리 | 공평성 | 수율 |

### Abstract

The existing buffer management scheme is caused by with burstness characteristic of the TCP traffic and with only the transmission which is not loss it provides the smallest transmission rate guarantee where the GFR demands and a fair characteristic.

In order to provide a high fair characteristic from the dissertation which it sees with the smallest transmission rate guarantee where the GFR demands it proposed the existing buffer algorithm which applies a Fuzzy mechanism in the existing buffer management technique. The proposed algorithm decides a packet disuse used by three parameters which are composed of tagging information, the buffer usage, and the load of VC. Simulation results shows that the fairness and goodput of the proposed algorithm were excellent where the size of MCR will become larger, from the switch the Double-EPD or the DFBA was visible a similar efficiency even from size change of the buffer. The algorithm which is proposed provides a good throughput and a fair characteristic.

■ keyword : | GFR | Fuzzy | Buffer Management | Fairness | Goodput |

---

## I. 서론

GFR 서비스는 UBR 서비스에 최소의 서비스 품질

보장을 해주기 위한 방식으로 UBR+라는 이름으로 제안되어 연구가 진행되다가 GFR로 바뀌어 현재 ATM Forum TM 4.1에서 표준화가 되었다[4].

GFR 서비스는 최대 패킷 크기를 초과하지 않는 최소 패킷레벨 보장을 위한 패킷 스트림을 지원한다. 또한 사용자에 의해 설정된 프레임의 우선순위 표시를 ATM 망에서 순응 검사 등을 통해 임의로 바꿀지의 여부에 따라 다시 GFR.1과 GFR.2로 구분된다[1][2]. GFR.1의 경우 초과 전달된 프레임을 순응 검사에서 폐기하도록 하며, GFR.2의 경우 CLP(Cell Loss Probability) 비트를 1로 태깅하여 프레임의 우선순위를 낮출 수 있도록 하고 있다[1,2]. GFR에서는 가용 대역폭에 대한 공평성 있는 대역할당을 지원해야 하기 때문에 버퍼관리를 통한 셀 폐기와 셀 스케줄링이 중요한 요소이다[3,4].

GFR 서비스가 요구하는 최소 전송률 보장과 공평성을 제공하기 위해서는 효율적인 버퍼 관리 알고리즘의 연구가 필수적이다. GFR의 요구사항을 만족시키기 위해 Double-EPD(Early Packet Discard), WFBA(Weighted Fair Buffer Allocation), DFBA(Differential Fair Buffer Allocation) 등의 버퍼 관리 알고리즘이 제안되었다[5-8].

기존의 버퍼 관리 알고리즘은 TCP 트래픽의 "burstness" 특성으로 인해 CLP가 0인 AAL-5 PDU의 손실 없는 전송만으로는 GFR이 요구하는 최소 전송률 보장과 공평성을 제공하기 어렵다. GFR의 대표적인 버퍼관리 알고리즘인 Double-EPD나 DFBA의 경우 버퍼의 셀 점유율이 임계치 이상일 때 셀 폐기를 수행하므로 임계치 이하에서 일부 VC가 할당된 버퍼공간보다 많은 셀을 버퍼에 받아들일 경우 공평성이 나빠지는 단점이 있다[9]. 또한 연결된 VC의 지연 시간이나 패킷의 크기가 서로 다른 경우에도 최소 전송률 보장과 공평한 대역 할당을 못하는 단점이 있다.

본 논문에서는 퍼지이론을 이용하여 GFR이 요구하는 최소 전송률 보장과 높은 공평성을 제공하기 위한 버퍼관리 알고리즘을 제안하였다. 제안된 알고리즘은 태깅정보, 전체 버퍼의 점유량, VC의 부하 등 세 개의 파라미터를 이용해 패킷 폐기를 결정한다. 전체 버퍼의 점유량은 현재 버퍼 사용율이 낮은 경우, 중간경우, 높은 경우의 세 가지 상태에 대한 멤버십 함수를 이용한다. VC의 부하는 연결된 각 VC에 따른 부하의 정도를

세 가지 상태를 이용해 입력 파라미터로 사용하였다. 이 세 가지 정보를 이용해 퍼지 알고리즘을 통해 패킷 폐기 파라미터를 구하고 이를 미리 정의된 셀 폐기 임계치와 비교하여 패킷을 폐기함으로써 MCR을 보장하고 수율을 높이며 잉여대역에 대해 공평성을 보장해 줄 수 있다.

## II. 버퍼관리 메커니즘

버퍼 관리는 버퍼에 들어오는 프레임의 수를 조절하기 위해 망의 스위치나 라우터에서 일반적으로 구현된다. 공통 버퍼를 여러 연결들이 공유해서 사용하는 방식(shared buffer)과, 각각의 연결에 대해 버퍼 점유량을 조절하는 연결 단위 버퍼 방식(per-connection buffer)이 있다. 연결 단위 버퍼 방식은 각 연결에 대한 버퍼 점유량을 알기 위해 각 연결 단위로 카운터를 둔다. 연결 단위 버퍼 방식의 예는 SD>Selective Drop)이나 FBA(Fair Buffer Allocation)방식이 있다. 연결 단위로 카운터를 두는 것은 오버헤드를 초래하지만, 비순응 프레임이 망의 입구에서 폐기되지 않는다면 각 연결에 대한 버퍼의 점유량을 조절하는 것이 매우 어려워지기 때문에 카운터를 사용하는 것이 바람직하다.

아래 [그림 1]은 공통 버퍼방식의 버퍼관리 제어를 보여주고 있다. 망에 들어오는 패킷은 일단 패킷의 헤더와 데이터를 분리해서 패킷 헤더는 버퍼 제어기에 패킷에 관한 정보를 제공하고 버퍼 제어기는 패킷의 헤더 정보에 기초하여 버퍼관리 알고리즘에 따라서 패킷을 전송할 것인지 폐기할 것인지를 결정하게 된다. 패킷의 데이터는 데이터 메모리에 저장되었다가 버퍼 제어기에서 결정된 전송 순서에 따라 패킷 헤더를 덧붙여 전송되게 된다. 버퍼 제어기는 패킷 헤더 정보와 데이터 메모리의 상태정보를 가지고 버퍼 관리 알고리즘에 따라 패킷 폐기와 패킷의 전송순서를 결정하여 데이터 메모리에 전송신호를 보냄으로써 데이터 메모리와 데이터 전송을 관리한다. 데이터의 전송을 효율적으로 하기 위해서는 버퍼 관리 알고리즘이 간략하면서도 패킷의 특성에 따른 전송을 수행할 수 있어야 한다.

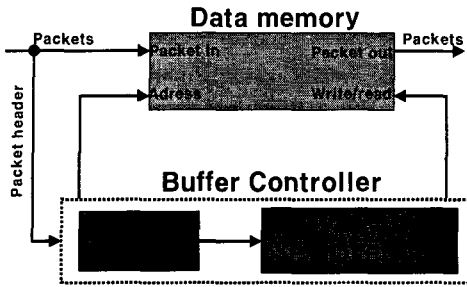


그림 1. 버퍼 관리 제어기

### III. 제안된 버퍼관리 알고리즘

#### 1. 제안된 퍼지 기반 버퍼관리 알고리즘

본 논문에서 제안된 퍼지 기반 버퍼 관리를 위한 알고리즘은 GFR 큐로부터 현재의 전체 사용율(Load of queue,  $Q_{occ}$ )과 각 VC별 사용율(Load of VC,  $VCi_{occ}$ )의 파라미터를 추출한다.

$$\delta_{cdf} = f_{Fuzzy}(Q_{occ}, VCi_{occ}) \quad (1)$$

TCP 소스들의 공평한 버퍼 할당을 위해 SD(select drop)를 사용한다. 즉, VC들의 공평한 버퍼 할당을 위해 각각의 활성 VC들의 버퍼 점유율을 추적하여 공평한 할당율보다 초과해서 사용하는 프레임은 선택적으로 폐기함으로써 서비스의 공평성을 보장하였다. [그림 2]는 제안된 알고리즘의 제어구조이며, 구성은 셀 폐기 파라미터 생성기와 비교기로 구성되어 있다.

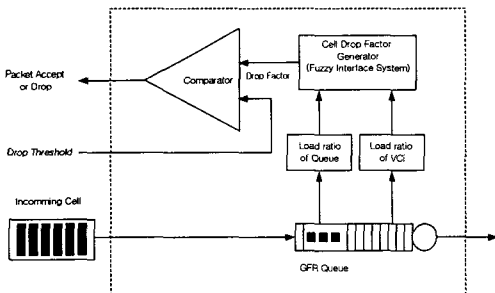


그림 2. 퍼지 기반 셀 제어기

#### 2. 퍼지 파라미터 생성

전체 큐에 상태  $Q_{occ}$ 를 세 부분으로 나누어 큐에 셀이 작은 구간, 중간 구간, 과잉구간으로 나누었다.  $Q_{occ}$ 를 [그림 3]에 나타내었다.

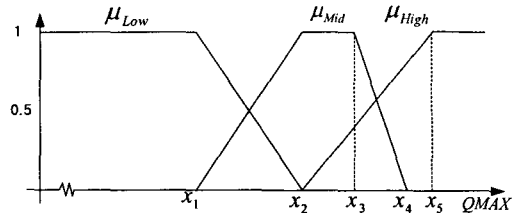


그림 3. 전체 큐 중 현재 상태변화량( $Q_{occ}$ )

이것은 다시 사다리꼴 퍼지수(Trapezium Fuzzy Number,  $Tr, F, N$ )로 나타낼 수 있다.

$$Q_{occ} = Tr.F.N \begin{cases} Low & (0, 0, x_1, x_2) \\ Mid & (x_1, x_2, x_3, x_4) \\ High & (x_2, x_5, 1, 1) \end{cases} \quad (2)$$

또, 각각의 멤버십 함수는

$$\mu_{Low}(x) = \begin{cases} 1 & (x \leq x_1) \\ 1 - \frac{x-x_1}{x_2-x_1} & (x_1 < x \leq x_2) \\ 0 & (x > x_2) \end{cases} \quad (3)$$

$$\mu_{Mid}(x) = \begin{cases} 0 & (x \leq x_1) \\ \frac{x-x_1}{x_2-x_1} & (x_1 < x \leq x_2) \\ 1 & (x_2 < x \leq x_3) \\ 1 - \frac{x-x_3}{x_4-x_3} & (x_3 < x \leq x_4) \\ 0 & (x > x_4) \end{cases} \quad (4)$$

$$\mu_{High}(x) = \begin{cases} 0 & (x \leq x_2) \\ \frac{x-x_2}{x_5-x_2} & (x_2 < x \leq x_5) \\ 1 & (x > x_5) \end{cases} \quad (5)$$

$$x_1 = \frac{MCR_i}{C_{GFR}} QT, \quad x_2 = \frac{MCR_i}{\sum MCR_i} QT \text{ 로 설정한}$$

다.  $C_{GFR}$ 은 GFR에 할당된 전체 대역이고,  $QT$ 는 큐에 총 셀수,  $MCR_i$ 는  $i$  번째 VC의 협상된  $MCR$  값이다. 여기서,  $x_1$ 은 셀 점유율이  $QT$ 일 때  $i$ -번째 VC의 최소 전송률에 따른 버퍼공간을 의미하고,  $x_2$ 는 셀 점유율이  $QT$ 일 때  $i$ 번째 VC에 공평히 할당된 대역에 관련되어 할당된 버퍼공간이다. 각각에 대한 멤버십 함수는,

$$\mu_{Low}(x) = \begin{cases} 1 & (x \leq x_1) \\ 1 - \frac{x-x_1}{x_2-x_1} & (x_1 < x \leq x_2) \\ 0 & (x > x_2) \end{cases} \quad (6)$$

$$\mu_{Mid}(x) = \begin{cases} 0 & (x \leq x_1) \\ \frac{x-x_1}{x_2-x_1} & (x_1 < x \leq x_2) \\ 1 - \frac{x-x_2}{0.5x_2} & (x_2 < x \leq 1.5x_2) \\ 0 & (x > 1.5x_2) \end{cases} \quad (7)$$

$$\mu_{High}(x) = \begin{cases} 0 & (x \leq x_2) \\ \frac{x-x_2}{0.5x_2} & (x_2 < x \leq 1.5x_2) \\ 1 & (x > 1.5x_2) \end{cases} \quad (8)$$

두 개의 입력을 가진 퍼지 시스템은 각각 3개의 멤버십 함수를 가지고 있으므로 모두 9개의 독립된 상태를 갖는다. 이 중에서 최적화된 7가지의 상태를 사용해서 [표 1]의 제어 규칙을 만들었다.

표 1. 퍼지 제어 규칙

if $Q_{occ}$	and $VC_i_{occ}$	then $\delta_{cdf}$ is
High	High	High
High	Medium or Low	Moderate_High
Medium	High	Moderate_High
Medium	Medium	Medium
Medium	Low	Moderate_Low
Low	High	Moderate_Low
Low	Medium or Low	Low

를 얻는다. 즉,  $Q_{occ} = \mu_{High}$  인 경우, 총 버퍼의 혼잡 상태이므로 새로운 패킷을 받지 않으며 현재의 연결 중  $CLP=1$  인 패킷과  $i$  번째 VC의 사용율이 일정 기준치 이상인 프레임은 삭제한다.  $Q_{occ} = \mu_{Mid}$  인 경우, 모든 연결은 서비스하며 새로운 연결 중  $CLP=1$  인 프레임은 폐기한다.  $Q_{occ} = \mu_{Low}$  인 경우, 모든 새로운 연결을 받아들인다.

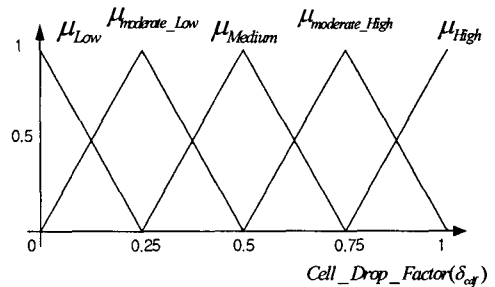


그림 4. 셀 폐기 파라미터( $\delta_{cdf}$ )

[그림 5]는 입력  $Q_{occ}$  및  $VC[i]_{occ}$ 에 대한  $\delta_{cdf}$  값을 3차원 그래프로 나타낸 것이다.

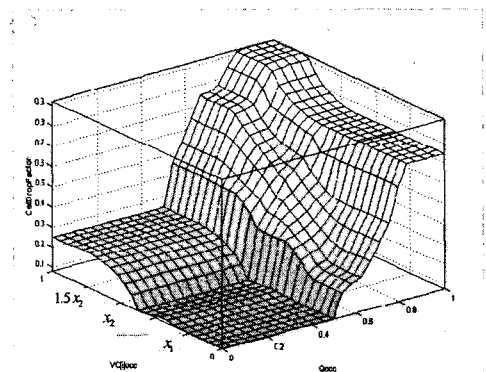


그림 5.  $VC[i]_{occ}$ ,  $Q_{occ}$ 와 Cell Drop Factor의 그래프

스위치에 프레임의 첫 번째 셀이 왔을 때, 먼저 셀 폐기 파라미터를 구하기 위한 입력값으로 전체 큐 사용을

위의 두 입력 값을 통해 [그림 4]의 셀 폐기 파라미터

( $Q_{occ}$ )과 현재 프레임에 해당하는  $VC_{occ}[i]$ 를 구한다. [그림 6]은 스위치에 셀이 들어왔을 때의 처리 과정을 속도 코드로 보여주고 있다.

```

when first cell of a frame arrived:
  Q_occ=LoadRatio_Queue(QT);
  VC_occ[i]=LoadRatio_VC(VCQ[i], MCR[i], QT);
  cdf=Generate_cdf(Q_occ, VC_occ); /* Generating cell drop factor */
  if(CLP_of_cell == 1 and cdf > Low_threshold){ /* There is not
  plenty
    of buffer space so even tagged cells can not be
    accepted */
    Drop_Cell;
    PS[i]=1; /* Make sure subsequent cells are discarded */
  }
  else if(CLP_of_cell == 0 and cdf > High_threshold){ /* Even
  though cell
    is not tagged, there is not space to accepting
    cells */
    Drop_Cell;
    PS[i]=1; /* Make sure subsequent cells are discarded */
  }
  else{ /* Otherwise */
    Accept_Cell;
    PS[i]=0; /* Remembers first cell was accepted */
    QT++; /* Increments total buffer content */
  }
when mid or last cell of a frame arrived:
  if(PS[i] == 1)
    Drop_Cell;
  else {
    Accept_Cell;
    QT++; /* Increments total buffer content */
  }
  
```

그림 6. 스위치에서 입력 셀 처리과정

#### IV. 시뮬레이션 및 성능 평가

본 논문에서 제안된 알고리즘의 시뮬레이션을 위한 네트워크 모델은 다른 알고리즘과의 비교를 위해, [그림 7]처럼 두 개의 ATM 스위치와 열 개의 TCP 소스들로 구성된[9]. 네트워크의 각 링크는 OC-1인 51.84 Mbps이며 ATM 스위치간의 지연은 10 msec이고 소스와 스위치간의 지연은 0.5 msec이다.

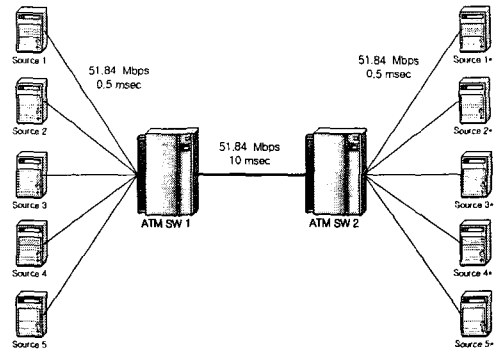


그림 7. 시뮬레이션을 위한 네트워크 모델

[그림 8]은 MCR의 총 크기를 30Mbps로 하여 시뮬레이션 한 결과이다. 여기서는 큰 MCR을 요구 할 때 제안된 버퍼 관리 알고리즘이 높은 성능을 보임을 알 수 있다. 반면에 Double-EPD의 경우는 거의 일정한 수율을 보임으로서 큰 MCR의 요구에 더욱 낮은 성능을 보였다.

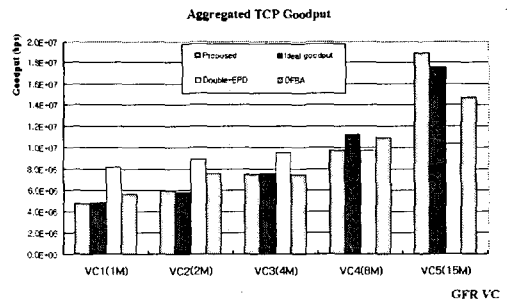


그림 8. TCP Goodput(MCR=30Mbps)

제안된 알고리즘의 경우 버퍼의 점유와 VC의 부하에 따라 Fuzzy 알고리즘에 의해 구한  $cdf$ 에 따라 폐기하여 공정성과 전체 TCP 수율이 DFBA나 Double-EPD보다 성능이 향상하였음을 보였다.

표 2. 성능비교(MCR=30Mbps)

Performance	Double-EPD	DFBA	Proposed
Fairness index	0.8943	0.9755	0.9952
Total Goodput	46.56Mbps	46.31Mbps	46.68Mbps

[그림 9]는 링크의 지연시간을 달리하여 시뮬레이션을 한 결과이다. 점선으로 나타낸 부분은 링크의 지연의 합이 11 msec로 설정하였으며, 실선으로 된 부분은 지연을 21 msec로 설정하여 시뮬레이션을 하였다. 지연이 적을 경우 적은 MCR에서 Double-EPD나 DFBA는 좋은 수율을 보여 주었으나 MCR이 클 경우나 지연 시간이 길어질 경우 성능이 급격히 떨어짐을 볼 수 있다. 두 경우 모두 제안된 알고리즘은 이상적인 수율에 근접하며 MCR과 지연에 거의 영향이 없음을 보여준다.

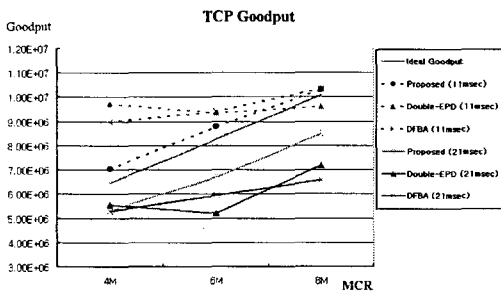


그림 9. 지연의 변화에 따른 성능 비교

### V. 결론

본 논문에서는 퍼지이론을 이용하여 GFR이 요구하는 최소 전송률과 공평성을 제공하기 위해서 퍼지 기반의 버퍼 관리 알고리즘을 제안했다. 제안된 알고리즘은 태깅 정보와 전체 버퍼의 사용율을 이용하여 스위치에서 버퍼상태에 따른 적절한 수용을 통하여 트래픽의 체증을 미리 회피할 뿐만 아니라 각 VC의 사용율에 따라 퍼지 알고리즘을 기반으로 새로운 패킷의 수용여부를 결정함으로써 MCR을 보장하고 잉여 대역에 대한 공평한 할당을 할 수 있었다. 시뮬레이션 결과 MCR의 크기가 커질수록 제안된 알고리즘의 수율과 공평성이 우수하였으며, 스위치에서의 버퍼의 크기 변화에서도 Double-EPD나 DFBA는 비슷한 성능을 보였고, 제안된 알고리즘은 좋은 수율과 높은 공평성을 제공하였다. 마지막으로 링크의 지연 시간의 합을 변화시켜 시뮬레이션을 한 결과 제안된 알고리즘의 경우 이상적인 수율에 근접하며 지연에 거의 영향이 없음을 보여 주었다.

### 참고 문헌

- [1] R. Goyal and Raj Jain, "Design Issues for providing Minimum Rate Guarantees to the ATM Unspecified Bit Rate Service," *Proceedings of the IEEE ATM Workshop Proceedings*, pp.169-175, 1998.
- [2] C. T. Chan, Y. C. Chen, and Pi-Chung Wang "An Efficient Traffic Control Approach for GFR Services in IP/ ATM Internetworks," *IEEE GLOBECOM 98*, Vol.2, pp.1112-1117, 1998.
- [3] O. Bonaventure and J. Nelissen "Guaranteed Frame Rate: A Better Service for TCP/IP in ATM Networks," *IEEE Network*, Vol.15, pp.46-54, JAN.-Feb. 2001.
- [4] S. Sathaye, "ATM Forum Traffic Management Specification, Version 4.0," *ATM Forum Technical Committee*, Mar. 1996.
- [5] J. Heinanen, K. Kilkki, "A fair buffer allocation scheme," Vol.21, pp.220-226, 1998.
- [6] R. Guerin and J. Heinanen, "UBR+ Service Category Definition," *ATM Forum contribution*, ATM96-1598, Dec. 1996.
- [7] O. Bonaventure. "Providing Bandwidth Guarantees to Internetwork Traffic in ATM Networks," *Proc. of IEEE ATM98 workshop*, May, 1998.
- [8] R. Goyal, R. Jain, S. Fahmy and B. Vandalore, "Buffer Management for the GFR Service," *ATM Forum/98-0405*.
- [9] V. Bonin, F. Cedan, "A Simulation Study of Differential Fair Buffer Allocation," *High Performance Switching and Routing, 2000. ATM 2000. Proceedings of the IEEE Conference on*, pp.365-372, 2000.

저 자 소 개

조 해 성(Hae-Seong Cho)

종신회원



- 2001년 2월 : 전북대학교 전자공학과 대학원 졸업(공학박사)
  - 2001년 9월~현재 : 건양대학교 전자정보공학과 교수
- <관심분야> : 멀티미디어 통신, 유비쿼터스 네트워킹, 네트워크 QoS