

IEEE 802.16e 네트워크 환경에서 ARIMA 트래픽 예측을 사용한 대역폭 프로비저닝

Bandwidth Provisioning Using ARIMA-Based Traffic Forecasting in IEEE 802.16e Networks

김 현 우 * 이 준 희** 최 용 훈*** 정 영 욱 **** 이 혁 준 *****
(Hyun-Woo Kim) (Jun-Hui Lee) (Yong-Hoon Choi) (Young-Uk Chung) (Hyun-Joon Lee)

요 약

본 논문에서는 IEEE 802.16e 네트워크 환경에서 기지국이 받은 트래픽을 예측하여 백본 네트워크에서의 대역폭 프로비저닝 방법을 제안한다. 트래픽은 4개의 클래스로 정의하고 각 클래스 별 트래픽을 박스-젠킨스의 시계열 분석법을 사용하여 예측하였다. 이를 바탕으로 프로비저닝을 진행하고 링크의 최대 대역폭을 초과하는 트래픽에 대해서 각 클래스의 우선순위를 최적으로 만족시키는 프로비저닝 방법을 제시하고 우선순위를 고려하지 않은 경우와 비교하였다. 또한 클래스 별 대역폭을 고정적으로 프로비저닝한 경우와 비교하여 패킷의 손실 및 QoS 보장률의 성능이 향상됨을 증명하였다.

Abstract

In this paper, we propose a dynamic bandwidth provisioning method based on traffic forecasting in IEEE 802.16e packet core network. The traffic is categorized as 4-different classes and the traffic amount of each class is forecasted by the Box-Jenkins method. To increase the service provider's revenue we provision the bandwidth of 4-different classes dynamically using greedy algorithm. The simulation results show that the number of packet drops is reduced and the level of QoS is improved compared with two different the methods - no priority considering and static provisioning.

Key words: IEEE 802.16e, traffic, forecasting, box-jenkins, ARIMA, provisioning, QoS

I. 서 론

국내에서는 와이브로라는 명칭으로 알려진 IEEE 802.16-2004 [1]와 IEEE 802.16e-2005 [2]기술은

60km/h 이상의 속도에서 이동성을 보장하는 이동 단말에게 1 Mbps 이상의 하향 링크 (DL, Down Link) 대역폭과 128 Kbps 이상의 상향 링크 (UL, Up Link) 대역폭 지원을 기본으로 하며, 끊김 없는 초고속 무

† 이 논문은 2007년 정부(교육인적자원부)의 재원으로 한국학술진흥재단의 지원을 받아 수행된 연구임 (KRF-2007-331-D00287)

* 주저자 : 광운대학교 제어계측공학과 박사

** 공저자 : 광운대학교 제어계측공학과 석사

*** 공저자 : 광운대학교 정보제어공학과 교수(교신저자)

**** 공저자 : 광운대학교 전자공학과 교수

***** 공저자 : 광운대학교 컴퓨터공학과 교수

† 논문접수일 : 2009년 1월 17일

† 논문심사일 : 2009년 2월 23일

† 게재확정일 : 2009년 2월 24일

선 인터넷 서비스를 제공할 수 있도록 설계되었다. 와이브로 기술에서는 단말의 QoS (Quality of Service) 를 보장하기 위해서 서비스 클래스를 구분하여 각 클래스별 차별된 스케줄링 방식을 사용하고 있지만 백본 네트워크 (backbone networks)에 대해서는 정의하고 있지 않다. 이러한 클래스별 QoS를 백본네트워크에서도 유지시켜주기 위해서 클래스별 대역폭 프로비저닝 (Bandwidth Provisioning) 기술이 필요하다.

일반적으로 대역폭 프로비저닝은 요구되는 대역폭보다 충분히 큰 양을 프로비저닝하는 over-provisioning 방식을 사용한다. 이는 혼잡이 발생하지 않고 지연이 적으며 시스템관리가 단순하다는 장점을 가지지만 대역폭이 낭비되는 단점이 있다. 또한 대역폭 프로비저닝 방식은 크게 Complete Partitioning 과 Complete Sharing 으로 나눌 수 있는데, 전자의 방식으로 제안된 연구로는 [3]가 있으며, 후자의 방식을 택한 연구로는 [4, 5]가 있다. 본 논문에서는 Complete Sharing 방식의 대역폭 프로비저닝을 따르고 있으며, Complete Partitioning 방식과 비교하여 실험을 진행하였다.

대역폭 프로비저닝에 대한 연구 가운데 Krithikaivasan et al. 은 트래픽 예측을 통해 사용할 대역폭을 프로비저닝하는 기술을 제안하였다. 이는 네트워크에 들어오는 트래픽의 양을 일정 시간 주기로 측정하여 시계열 데이터로 변환하고 이러한 샘플데이터를 ARIMA 모델을 기초로 네트워크 환경에 맞는 예측을 하여, 그 양을 동적으로 프로비저닝 하는 기술이다. Krithikaivasan et al. [6, 7]이 제안한 방안은 트래픽 예측의 정확성은 높지만 클래스별 분류를 하지 않고 전체 트래픽 양을 예측하여 프로비저닝 하기 때문에 클래스별 QoS 는 보장되지 않는다는 단점이 있다. 또한 IEEE 802.16e 네트워크 환경에서의 프로비저닝 방식에 관한 연구로 [4, 5]가 제안되었다.

Al-Manthari et al., [4]의 방안은 WiMAX와 같은 광대역 무선 네트워크 환경에서 다양한 트래픽 클래스를 고려한 방법으로 일정한 주기마다 동적으로 대역폭을 프로비저닝하는 기술이다. 이때 프로비저닝하는 양은 이전 시점의 트래픽양에 기초하므로 대역폭의 부족 및 낭비가 커지는 단점을 지닌다.

Khemiri et al., [5]의 방안은 WiMAX 환경에서 CAC (call admission control)를 기반으로 각 클래스별 우선순위를 고려하여 프로비저닝하는 기술이다. 이 방식은 호 단위로 QoS를 관리하기 때문에 대용량 데이터 서비스 위주의 WiMAX환경에는 적합하지 않다는 단점을 갖는다.

따라서 본 논문에서는 클래스별 QoS를 보장하기 위해 IEEE 802.16e 기술에서 정의하는 서비스 클래스별로 트래픽을 분류하고 ARIMA 모델을 사용하여 예측하였으며, Fractional Knapsack Problem 및 Greedy algorithm 을 사용하여 대역폭 프로비저닝을 최적화 시켜 클래스 간 우선순위를 보장하는 기술을 제안한다.

본 논문은 다음과 같이 구성된다. 2장에서는 IEEE802.16e 기술에서의 QoS보장 기술을 비롯하여 기존의 IEEE802.16e 환경에서의 대역폭 프로비저닝 방식과 트래픽 예측을 사용한 대역폭 프로비저닝 방식에 대해 소개한다. 3장에서는 ARIMA 모델에 대한 설명과 제안하는 클래스별 대역폭 프로비저닝 알고리즘을 설명하며 4장에서는 제안하는 방안의 성능을 평가하고 5장에서 결론을 맺는다.

II. 관련 연구

IEEE 802.16e 기술은 단말의 QoS를 보장하기 위해 상향링크(UL, Up Link) 스케줄링에 대해서 <표 1>과 같이 5가지의 서비스 클래스를 정의하고 있다 [2].

UGS (Unsolicited Grant Service)는 silence suppression 이 없는 VoIP 처럼 주기적으로 일정 크기의 데이터 패킷을 발생시키는 실시간성 트래픽을 지원하기 위

<표 1> IEEE 802.16e 기술에서의 서비스 클래스
<Table 1> The service class at the IEEE 802.16e

Service class	Application
UGS	T1, E1, VoIP(without silence suppression)
rtPS	Streaming Video
ertPS	VoIP(with silence suppression)
nrtPS	FTP
BE	e-mail

한 스케줄링 방식이며 rtPS(Real Time Polling Service)는 MPEG(Moving Pictures Group)비디오처럼 가변 크기의 실시간 트래픽에 대해 대역폭을 할당하는 스케줄링 방식이다. ertPS (Extended Real Time Polling Service)는 UGS와 rtPS의 효율성을 모은 스케줄링 방식으로 silence suppression을 포함하는 VoIP를 지원한다. nrtPS (non-Real Time Polling Service)는 파일전송(File transfer)과 같은 가변크기의 비실시간적인 트래픽을 지원하며, BE (Best Effort)는 e-mail과 같은 일반적인 최선형 트래픽에 대해서 단말이 경쟁기반의 요청기회를 사용할 수 있도록 허락하는 방식이다.

이러한 IEEE802.16e 환경에서 QoS를 보장하기 위해서 [4, 5]에서는 대역폭 프로비저닝 방법을 제안하고 있다.

[4]에서는 WiMAX 와 같은 광대역 무선 네트워크 환경에서 서로 다른 대역폭을 요구하는 다양한 트래픽 클래스에 대한 QoS를 보장하기 위해서 동적인 대역폭 프로비저닝 방식을 제안한다. 시점 t 에서 다음 SL frames 이후의 대역폭을 프로비저닝한다. 이때 프로비저닝하는 양은 SL frames 전의 트래픽을 토대로 클래스별 우선순위에 따라 가중치를 부여하여 결정한다.

[5]에서는 WiMAX환경에서 CAC 기반으로 QoS를 관리하는 방법을 제안한다. 요청되는 Call의 수락 여부를 결정하는 정책은 Priority-based bandwidth sharing strategy 를 사용하여 트래픽의 클래스별 우선순위를 보장하며 CSMDP (Constrained Semi-Markov Decision Processes) 를 사용하여 모델링 하였다.

트래픽을 예측하여 대역폭을 프로비저닝하는 기술은 [6, 7]을 통해서 연구가 되었다. [6]은 기지국 또는 라우터에서 받은 전체 트래픽을 ARIMA 모델을 사용하여 예측하고 동적으로 프로비저닝 하였다. Probability-hop forecast algorithm을 사용하여 예측의 정확성을 높여 최소한의 패킷손실을 얻어 유저의 QoS를 보장하려고 하였다. 또한 예측의 결과를 토대로 IBP (Instantaneous Bandwidth Provisioning), SBP (Stabilized Bandwidth Provisioning), SBPL (Stabilized Bandwidth Provisioning with Local Maxima)을 각각 적용하여 성능평가를 진행하였다. 또한 [7]의 연구는 [6]의 연구에

ARCH (Auto Regressive Conditional Heteroskedasticity) 모델을 추가적으로 사용하여 예측을 진행하였다.

III. ARIMA 트래픽 예측을 사용한 대역폭 프로비저닝 기술

본 장에서는 트래픽을 클래스 별로 분류하고 이를 박스-젠킨스의 시계열분석법을 사용하여 예측한 후 이를 토대로 한 동적 대역폭 프로비저닝 기술을 서술한다. 또한 링크의 최대 대역폭을 초과하는 경우에 대해서 Fractional Knapsack Problem을 적용하여 문제를 정의하고 이를 통해 최적의 해를 찾아내는 방법을 설명한다.

트래픽을 예측하기 위해서 본 논문에서는 박스-젠킨스의 시계열 분석법을 사용하였다. 박스-젠킨스의 시계열 분석법은 시계열은 현재의 상태가 과거 및 미래의 상태와 밀접한 관계를 갖고 있으며 시간의 흐름에 독립이 아닌 자기 상관관계를 갖는다는 점 [8, 9] 에 착안한다.

비정상 시계열 $\{X_t\}$ 에서, 음이 아닌 정수 d 에 대한 차분계열 $(1-B)^d X_t$ 가 ARMA(p,q) 과정을 따르면 $\{X_t\}$ 를 차수 (p,d,q) 인 자기회귀 누적 이동평균과정 (autoregressive integrated moving average process of order (p,d,q))라 하고 ARIMA(p,d,q)로 표시한다.

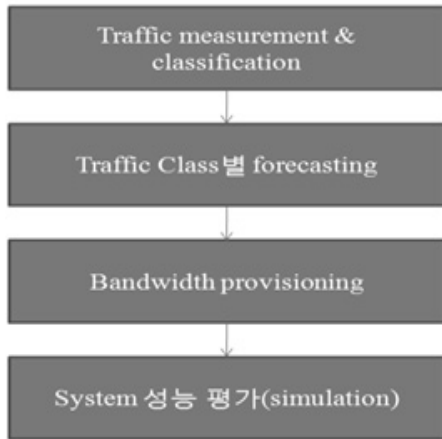
또한 ARMA(p,q)는 식 (1)과 같이 표현할 수 있다.

$$X_t + \alpha_1 X_{t-1} + \alpha_2 X_{t-2} + \dots + \alpha_p X_{t-p} = \beta_0 e_t + \beta_1 e_{t-1} + \dots + \beta_q e_{t-q} \quad (1)$$

NLANR [10] 에서 제공하는 24시간 동안 수집한 트래픽 측정자료를 6분 간격으로 재 샘플링하여 데이터셋으로 사용하였고 well-known port number를 사용하여 4가지 클래스로 분류하였다.

각 클래스별 데이터는 처음 80개의 데이터를 샘플로 ARIMA모델의 차수를 결정하고 SPSS12.0을 사용하여 식 (1)의 α, β 를 추정하였다.

추정된 \hat{X}_t 를 사용하여 81번째 데이터를 예측하고, 81번째 실제값을 포함한 이전 80개의 데이터를 바탕으로 α, β 를 다시 추정하여 나온 \hat{X}_t 를 사용하여 82



<그림 1> 제안하는 프로비저닝 기술 개요

<Fig. 1> Summary of proposal provisioning technology

번째의 데이터를 예측하였다. 이를 반복하여 24시간의 트래픽을 예측하였고, 예측 결과는 4장에서 확인할 수 있다.

<그림 1>과 같이 트래픽을 측정하고 클래스 별로 분류한 후 각 클래스별로 예측된 결과를 바탕으로 대역폭을 동적으로 프로비저닝 한다. 이 경우 각 클래스별 예측결과와 합이 전체 허용 대역폭을 초과할 경우 그 중 일부는 버리고 프로비저닝 해야 한다. 이를 위해 대역폭 할당을 Fractional Knapsack Problem으로 정의하고 Greedy algorithm을 사용해 최적해를 구했다.

Knapsack Problem이란 정해진 용량의 Knapsack이 있을 때 이 Knapsack에 주어진 물건을 넣어서 최대한의 가치를 내는 문제이다. 각 물건은 고유한 무게와 가치가 있으며 Knapsack에 들어가는 모든 물건의 무게의 합은 Knapsack 용량을 초과하지 못한다.

Fractional Knapsack Problem은 물건을 쪼갤 수 있는 경우이며 물건의 무게 당 가치를 구하여 Greedy 방법으로 풀 수 있다 [11].

본 논문에서는 Fractional Knapsack Problem을 사용하였으며 물건의 무게 당 가치(value/weight)를 계산하고 그 크기 순서대로 채워 넣음으로써 최적해를 계산하였다.

또한 Knapsack 용량, 물건, 물건의 무게, 물건의 가치는 <표 2>와 같이 정의하였다.

<표 2> Fractional Knapsack Problem의 변수
<Table 2> Variables of Fractional Knapsack Problem

Knapsack 용량	링크의 최대 대역폭
물건(item)	각 서비스 클래스 (BE, nrtPS, rtPS, UGS)
물건의 무게(weight)	각 서비스 클래스별 예측된 트래픽 양
물건의 가치(value)비	BE:nrtPS:rtPS:UGS = 1:2:3:4

알고리즘

```

a = 0 ;
1:  $F_{t+1}(BE) \leftarrow \hat{X}_{t+1}(BE)$ 
2:  $F_{t+1}(nrtPS) \leftarrow \hat{X}_{t+1}(nrtPS)$ 
3:  $F_{t+1}(rtPS) \leftarrow \hat{X}_{t+1}(rtPS)$ 
4:  $F_{t+1}(UGS) \leftarrow \hat{X}_{t+1}(UGS)$ 
5: if  $F_{t+1}(BE) + F_{t+1}(nrtPS) + F_{t+1}(rtPS) + F_{t+1}(UGS) < MaxSize$  then
6:
 $PV_{t+1}(1) \leftarrow F_{t+1}(BE)$ 
•  $\frac{MaxSize}{F_{t+1}(BE) + F_{t+1}(nrtPS) + F_{t+1}(rtPS) + F_{t+1}(UGS)}$ ;
7:
 $PV_{t+1}(2) \leftarrow F_{t+1}(nrtPS)$ 
•  $\frac{MaxSize}{F_{t+1}(BE) + F_{t+1}(nrtPS) + F_{t+1}(rtPS) + F_{t+1}(UGS)}$ ;
8:
 $PV_{t+1}(3) \leftarrow F_{t+1}(rtPS)$ 
•  $\frac{MaxSize}{F_{t+1}(BE) + F_{t+1}(nrtPS) + F_{t+1}(rtPS) + F_{t+1}(UGS)}$ ;
9:
 $PV_{t+1}(4) \leftarrow F_{t+1}(UGS)$ 
•  $\frac{MaxSize}{F_{t+1}(BE) + F_{t+1}(nrtPS) + F_{t+1}(rtPS) + F_{t+1}(UGS)}$ ;
10: else
11:  $BE \leftarrow \frac{Value(BE)}{F_{t+1}(BE)}$ ;
12:  $nrtPS \leftarrow \frac{Value(nrtPS)}{F_{t+1}(nrtPS)}$ ;
13:  $rtPS \leftarrow \frac{Value(rtPS)}{F_{t+1}(rtPS)}$ ;
14:  $UGS \leftarrow \frac{Value(UGS)}{F_{t+1}(UGS)}$ ;
15:  $p1, p2, p3, p4 \leftarrow$  Variant Bubble Sort (BE, nrtPS, rtPS, UGS);
16: if  $F_{t+1}(p1) < MaxSize$  then
    
```

```

17:       $PV_{t+1}(1) \leftarrow F_{t+1}(p1);$ 
18:       $MaxSize \leftarrow MaxSize - PV_{t+1}(1);$ 
19:  else
20:       $PV_{t+1}(1) \leftarrow MaxSize;$ 
21:       $MaxSize \leftarrow 0;$ 
22:       $a \leftarrow 1;$ 
23:  if  $a=0$  and  $F_{t+1}(p2) < MaxSize$  then
24:       $PV_{t+1}(2) \leftarrow F_{t+1}(p2);$ 
25:       $MaxSize \leftarrow MaxSize - PV_{t+1}(2);$ 
26:  else if  $a=0$  then
27:       $PV_{t+1}(2) \leftarrow MaxSize;$ 
28:       $MaxSize \leftarrow 0;$ 
29:       $a \leftarrow 1;$ 
30:  if  $a=0$  and  $F_{t+1}(p3) < MaxSize$  then
31:       $PV_{t+1}(3) \leftarrow F_{t+1}(p3);$ 
32:       $MaxSize \leftarrow MaxSize - PV_{t+1}(3);$ 
33:  else if  $a=0$  then
34:       $PV_{t+1}(3) \leftarrow MaxSize;$ 
35:       $MaxSize \leftarrow 0;$ 
36:       $a \leftarrow 1;$ 
37:  if  $a=0$  and  $F_{t+1}(p4) < MaxSize$  then
38:       $PV_{t+1}(4) \leftarrow F_{t+1}(p4);$ 
39:       $MaxSize \leftarrow MaxSize - PV_{t+1}(4);$ 
40:  else if  $a=0$  then
41:       $PV_{t+1}(4) \leftarrow MaxSize;$ 
42:       $MaxSize \leftarrow 0;$ 
43:       $a \leftarrow 1;$ 
    
```

1~4는 각 클래스 별로 ARIMA모델을 사용해 추정된 결과를 예측값으로 사용함을 의미한다.

MaxSize는 링크의 최대 대역폭을 의미하며, 각 클래스 별로 예측된 값의 합이 이보다 작을 경우 6~9와 같이 프로비저닝 한다. 이 경우 각 예측된 값에 $\frac{MaxSize}{\text{예측된값의합}}$ 을 곱하여 링크의 최대 대역폭을 다 사용할 수 있는 프로비저닝을 한다. PV_{t+1} 는 각 클래스 별 프로비저닝 양을 의미한다.

또한 각 클래스 별로 예측된 값의 합이 링크의 최대 대역폭인 MaxSize보다 클 경우 트래픽의 손실은 필수적이다. 따라서 Fractional Knapsack Problem 으로 문제를 정의하여 최적의 프로비저닝을 시도하였다. 우선 11~14 와 같이 각 클래스 별 무게당 가치 (value/weight)를 계산하였으며 각 클래스 별 weight 및 value는 <표 2>와 같이 정의하였다. 클래스 별 무

게당 가치는 크기 별로 프로비저닝의 순서가 되기 때문에 15에서와 같이 내림차순으로 정렬하였다. 그 후 계산된 무게당 가치가 가장 큰 클래스의 예측값을 먼저 프로비저닝하고, 남은 대역폭이 있을 경우 다음 무게당 가치의 클래스를 프로비저닝하는 Greedy 알고리즘은 16-43 에서 볼 수 있다.

IV. 모의 실험 및 고찰

본 장에서는 클래스별 트래픽을 ARIMA 모델을 사용하여 예측된 결과와 이를 토대로 대역폭 프로비저닝 한 결과를 보여준다. 또한 클래스별 고정적인 대역폭 프로비저닝 방식, 클래스별 우선순위를 고려하지 않고 프로비저닝 한 결과와 비교하여 성능분석을 하였다. 또한 실험을 위한 트래픽은 NLNR 에서 제공하는 데이터를 사용하였으며, 2006년 4월 23일, 2006년 4월 30일 의 각 24시간 수집한 데이터이다.

트래픽 분류(classification)는 well-known port number 를 토대로 <표 3>과 같이 분류하였고 2분 간격으로 샘플링 된 데이터를 6분 간격으로 재구성하였다.

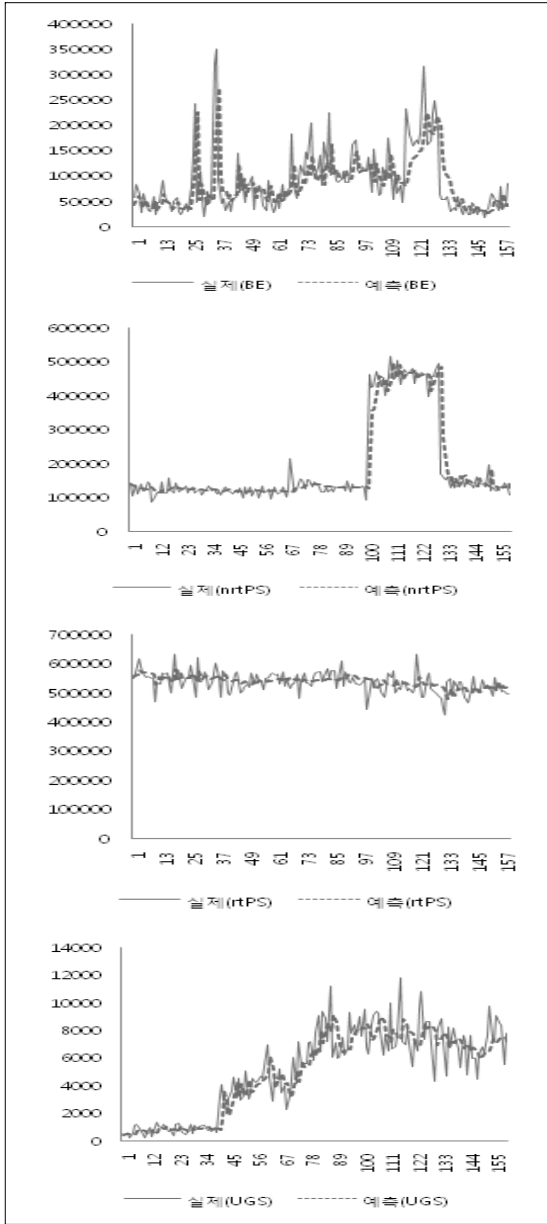
각 클래스별 처음 80개의 데이터를 샘플로 ARIMA모델의 차수를 결정하고 time lag=1로 예측을 진행하였다.

1. 2006년 4월 23일 샘플 데이터의 예측결과

2006년 4월 23일 데이터를 <표 3>의 포트번호를 이용하여 클래스 별로 분류하고, 각 클래스 별로 처음 80개의 데이터를 샘플로 하여 ARIMA모델의 차수를 결정하여 예측한 결과는 <그림 2>와 같다. 샘플데이터의 ACF(Auto Correlation Function)와 PACF

<표 3> 포트번호를 사용한 클래스별 트래픽 분류
<Table 3> Traffic classification of each class using the port number

Class	BE	nrtPS	rtPS	UGS
Application	E-mail	FTP	Video Streaming	VoIP
Port number	25,80	20,21,53,2122,3830,6346,6881,9001,9875	6660,6661	1863



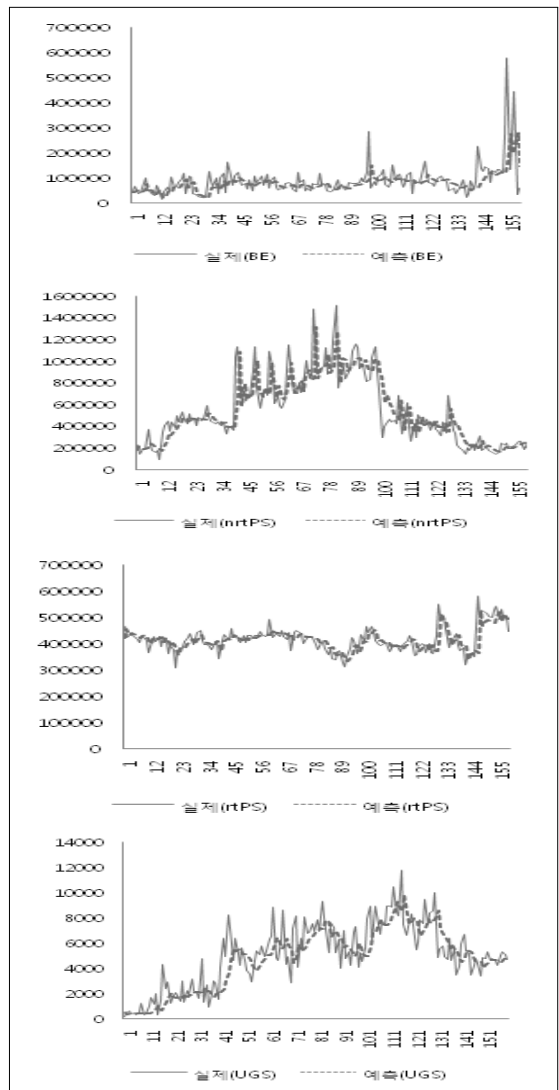
<그림 2> 2006년 4월 23일 데이터의 예측결과
<Fig. 2> Prediction of the data on April 23, 2006

(Partial Auto Correlation Function)를 계산한 결과 BE, nrtPS, UGS데이터는 1차분 결과의ACF와 PACF를 다 시차 2부터 0으로 수렴하여ARIMA(1,1,1), rtPS데이터의 경우 1차분 결과의ACF는 시차2부터 0으로 수렴하고, 1차분 결과의PACF는 시차3부터 0으로 수렴하

여ARIMA(2,1,1)을 적용하였다. 각 데이터의 ACF와 PACF, 모수의 추정은 SPSS12.0을 사용하여 진행하였다. [12]

2. 2006년 4월 30일 샘플 데이터 예측결과

<그림 3>은 2006년 4월 30일 데이터의 예측결과이다. 각 클래스 별 데이터는 처음 80개를 샘플로 ACF와 PACF를 계산하여 ARIMA모델의 차수를 결



<그림 3> 2006년 4월 30일 데이터의 예측결과
<Fig. 3> Prediction of the data on April 30, 2006

정하였다. BE 데이터는 샘플데이터의 ACF와 PACF가 시차2부터 0으로 수렴하여 ARIMA(1,0,1)을 적용하였다. nrtPS 데이터는 1차분 결과의 ACF와 PACF가 시차 3부터 0으로 수렴하여 ARIMA(2,1,2), rtPS, UGS 데이터는 1차분 결과의 ACF와 PACF가 시차 2부터 0으로 수렴하여 ARIMA(1,1,1)을 적용하였다.

3. 프로비저닝 결과

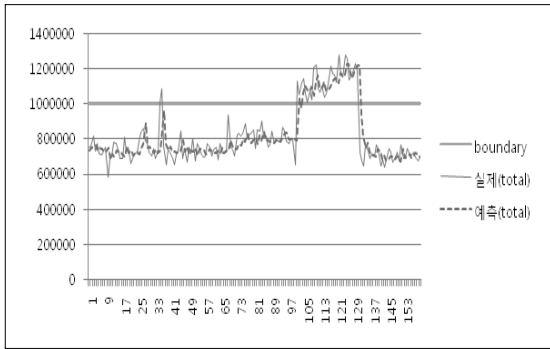
실험을 위해서 링크 최대 대역폭(MaxSize)은 <그림 4, 5>와 같이 2006년 4월 23일 데이터는 1(Mbytes), 2006년 4월 30일 데이터는 1.3(Mbytes)로 정의하였다. <그림 6, 7>은 프로비저닝에 따른 패킷손실을 보

여준다.

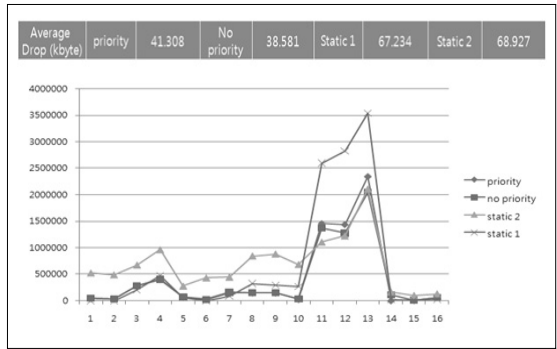
priority는 제안하는 방법의 결과이고 no priority는 예측결과를 바탕으로 동적인 프로비저닝을 하지만 클래스간 우선순위를 고려하지 않은 경우이고 static1과 static2는 고정적인 프로비저닝을 하였을 때 결과이다.

static1은 각 클래스별 트래픽의 평균을 고정적으로 프로비저닝하였고 static2는 각 클래스별 peak 트래픽의 전후 5시점의 평균에 각 클래스별 가중치를 부여하여 고정적인 프로비저닝을 하였다. 클래스별 가중치는 nrtPS 트래픽은 15% rtPS 트래픽은 20% UGS 트래픽은 25%로 하였다.

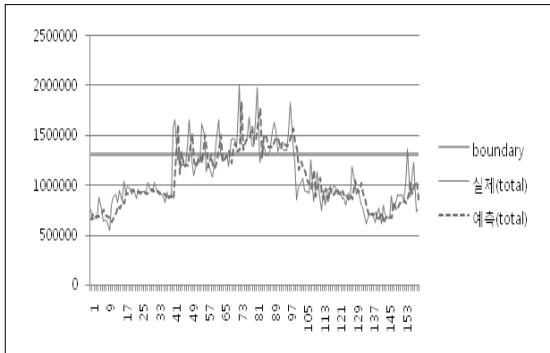
2006년 4월 23일, 30일 데이터 모두 고정적인 프



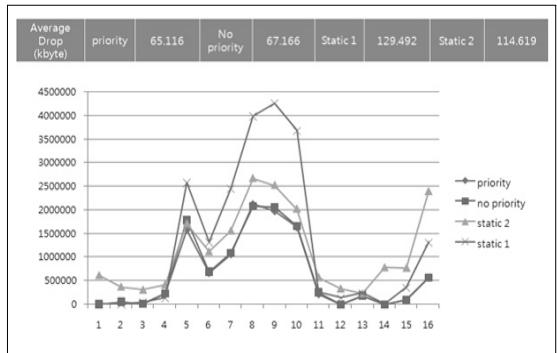
<그림 4> 2006년 4월23일 데이터의 클래스별 트래픽 예측결과와의 합
<Fig. 4> Sum about the prediction of each class on April 23, 2006



<그림 6> 2006년 4월23일 데이터의 프로비저닝에 따른 패킷손실 (단위:bytes)
<Fig. 6> Packet drop according to the provisioning of the data on April 23, 2006 (unit:bytes)



<그림 5> 2006년 4월30일 데이터의 클래스별 트래픽 예측결과와의 합
<Fig. 5> Sum about the prediction of each class on April 30, 2006



<그림 7> 2006년 4월30일 데이터의 프로비저닝에 따른 패킷손실 (단위:bytes)
<Fig. 7> The packet drop according to the provisioning of the data on April 30, 2006 (unit:bytes)

로비저닝(static1,2)에 비해서 ARIMA모형을 사용하여 예측된 결과를 바탕으로 동적인 프로비저닝(priority, no priority)한 경우 평균 패킷손실이 작음을 알 수 있다.

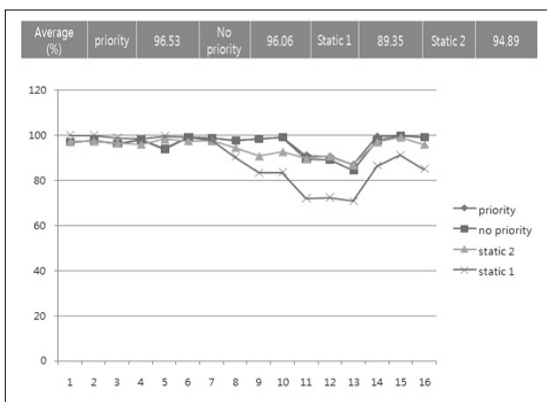
4. 우선순위를 고려한 QoS 보장률

(3)절의 패킷손실은 클래스별 손실의 합을 의미하므로 낮은 우선순위의 클래스의 패킷손실이 클 경우 전체 패킷손실도 커지는 경향이 있다. 따라서 클래스별 우선순위를 고려하여 패킷손실의 비율을 식(2), (3)과 같이 계산하였다

$$drop\ ratio = \min\left(\frac{Provisioning}{Revd\ traffic}, 1\right) \quad (2)$$

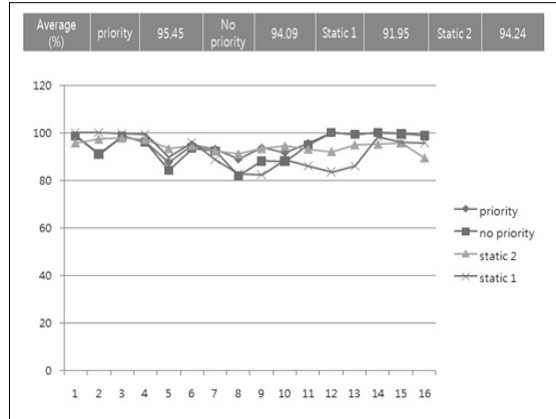
$$\begin{aligned} & \text{우선순위를 고려한 QoS보장률(\%)} \\ & = (BE\ drop\ ratio \cdot 10) + (nrtPS\ drop\ ratio \cdot 20) \\ & + (rtPS\ drop\ ratio \cdot 30) + (UGS\ drop\ ratio \cdot 40) \end{aligned} \quad (3)$$

drop ratio는 프로비저닝에 따른 패킷손실의 비율을 의미하며, 우선순위를 고려한 QoS 보장률은 각 클래스별로 계산한 drop ratio에 클래스별 가중치를 주어 계산하였다. 클래스별 가중치는 BE : nrtPS : rtPS : UGS = 1 : 2 : 3 : 4로 정의하였다. <그림 8, 9>와 같이 2006년 4월 23일, 30일 데이터 모두 제안하는 방



<그림 8> 2006년 4월 23일 데이터의 우선순위를 고려한 QoS 보장률

<Fig. 8> QoS guarantee rate in which it considers the priority of the data on April 23, 2006



<그림 9> 2006년 4월 30일 데이터의 우선순위를 고려한 QoS 보장률

<Fig. 9> QoS guarantee rate in which it considers the priority of the data on April 30, 2006

식(priority)이 가장 우수한 성능결과를 보여줌을 알 수 있다.

V. 결 론

IEEE 802.16e 기술은 서비스 클래스를 정의하고 단말과 기지국사이의 QoS를 보장하고 있다. 하지만 표준에서는 백본 네트워크에 대한 QoS는 정의하고 있지 않다. 본 논문의 목적은 IEEE 802.16e에서 제공되는 서비스 클래스별 QoS를 백본 네트워크에서도 유지 시키는 것이다. 제안하는 방법으로는 기지국에서 받은 트래픽을 IEEE 802.16e에서 정의하는 서비스 클래스로 나누고 각 클래스 별로 예측을 하여 대역폭을 동적으로 프로비저닝 한다.

예측방법으로는 박스-젠킨스의 ARIMA 모델을 사용하였다. 이러한 방법은 기존의 고정적인 대역폭 프로비저닝 기술과 비교하여 그 성능이 우수해짐을 보여 주었다. 또한 클래스별 우선순위를 고려하여 Fractional Knapsack Problem으로 문제를 정의하고 최적해를 구함으로써 프로비저닝 하였다. 이는 클래스별 우선순위를 고려하지 않은 경우와 비교해 IEEE 802.16e에서 제공되는 서비스 클래스별 QoS를 보장하고 그 성능이 향상됨을 보여주었다.

참고 문헌

- [1] IEEE Std. 802.16-2004, *IEEE Standard for Local and metropolitan area networks, Part 16: Air Interface for Fixed Broadband Wireless Access System*, Oct. 2004
- [2] IEEE Std. 802.16e-2005, *IEEE Standard for Local and metropolitan area networks, Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access System*, Dec. 2005
- [3] D. Niyato and E. Hossain, "A queuing-theoretic and optimization-based model for radio resource management in IEEE 802.16 broadband wireless networks," *IEEE Trans. Computers*, vol. 55, no. 11, pp. 1473-1488, Nov. 2006.
- [4] B. Al-Manthari, N. A. Ali, N. Nasser, and H. Hassanein, "Dynamic bandwidth provisioning with fairness and revenue considerations for broadband wireless communication," *Proc. IEEE Int. Conf. Communications*, pp. 4028-4032, May 2006.
- [5] S. Khemiri, K. Boussetta, N. Achir, and G. Pujolle, "WiMax bandwidth provisioning service to residential customers," *Proc. 9th Int. Conf. Mobile and Wireless Communications Networks*, pp. 19-21, Sept. 2007.
- [6] B. Krithikaivasan, K. Deka, and D. Medhi, "Adaptive bandwidth provisioning envelope based on discrete temporal network measurements," *IEEE INFOCOM*, vol. 3, pp. 1786-1796, Mar. 2004.
- [7] B. Krithikaivasan, K. Deka, and D. Medhi, "ARCH-based traffic forecasting and dynamic bandwidth provisioning for periodically measured nonstationary traffic," *IEEE Trans. Networking*, vol. 15, no. 3, pp. 683-696 June 2007.
- [8] G. E. P. Box, G. M. Jenkins, and G. C. Reinsel, *Time Series Analysis*, Prentice-Hall, Feb. 1994.
- [9] 김해경, 김태수, *시계열 분석과 예측이론*, 경문사, 2003. 8.
- [10] NLANR, <http://www.nlanr.net/>
- [11] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, The MIT Press, Jan. 2001.
- [12] 박성현, 조신섭, 김성수, *한글 SPSS*, 한나래, 2006. 9.

저자소개



김 현 우 (Kim, Hyun-Woo)

2002년 3월 ~ 2006년 2월 : 광운대학교 정보제어공학 학사
 2007년 3월 ~ 2009년 2월 : 광운대학교 제어계측공학 석사
 2009년 3월 ~ 현재 : 광운대학교 제어계측공학 박사과정



이 준 회 (Lee, Jun-Hui)

2002년 3월 ~ 2008년 2월 : 광운대학교 정보제어공학 학사
 2008년 3월 ~ 현재 : 광운대학교 제어계측공학 석사과정



최 용 훈 (Choi, Yong-Hoon)

1991년 3월 ~ 1995년 2월 : 연세대학교 전자공학 학사
1995년 3월 ~ 1997년 2월 : 연세대학교 전자공학 석사
1997년 1월 ~ 2001년 2월 : 연세대학교 전기전자공학 박사
2001년 4월 ~ 2002년 3월 : University of Maryland, Postdoctoral Research Associate
2002년 6월 ~ 2005년 8월 : LG전자 책임연구원
2005년 9월 ~ 현재 : 광운대학교 정보제어공학과 교수



정 영 옥 (Chung, Young-Uk)

1993년 3월 ~ 1997년 2월 : 한국과학기술원 전자전산학 학사
1997년 3월 ~ 1999년 2월 : 한국과학기술원 전기 및 전자공학 석사
1999년 1월 ~ 2003년 2월 : 한국과학기술원 전기 및 전자공학 박사
2003년 3월 ~ 2005년 8월 : 하이닉스 반도체 책임연구원
2005년 9월 ~ 현재 : 광운대학교 전자공학과 조교수



이 혁 준 (Lee, Hyukjoon)

1985년 9월 ~ 1987년 8월 : University of Michigan, Ann Arbor (학사)
1987년 9월 ~ 1989년 6월 : 시라큐스대학교 전산학 석사
1990년 1월 ~ 1993년 12월 : 시라큐스대학교 전산학 박사
1994년 3월 ~ 1996년 2월 : (주) 삼성전자 멀티미디어연구소 선임연구원
1996년 3월 ~ 현재 : 광운대학교 컴퓨터공학과 교수