
인벤토리모델을 이용한 인지무선네트워크에서 스펙트럼풀링

변상선

부산가톨릭대학교

An Inventory Model-based Spectrum Pooling in Cognitive Radio Networks

Sang-Seon Byun

Catholic University of Pusan

E-mail : ssbyun@cup.ac.kr

요 약

인지무선 (cognitive radio) 환경에서 2차사용자 (secondary user) 들의 일시적인 서비스 요구를 효과적으로 다룰 수 있는 스펙트럼풀링 (spectrum pooling) 기법이 최근 주목을 받고 있다. 스펙트럼풀링은 WSP (wireless service provider) 에 의해서 관리되며, 사용자의 요구가 있을 때, 금전적인 지불을 받고 스펙트럼을 2차사용자에게 대여해 주는 방식이다. 여기서, WSP는 가급적 많은 이윤을 남기기를 원한다. 이 논문에서 우리는 WSP가 유지하는 스펙트럼 풀 (pool) 을 확률 인벤토리모델 (probabilistic inventory model) 로 표현하고 2차사용자의 스펙트럼 요구가 정규분포를 따를 때, WSP가 비용을 최소화시키는 방향으로 인벤토리를 운영하는 전략을 제시한다. WSP가 지불해야하는 비용에는 일반적인 인벤토리 모델과 마찬가지로, 주문비용, 보관비용, 스톡아웃비용이다. 우리는 시뮬레이션을 통해, 고정된 인벤토리 레벨을 유지하는 것보다 확률 인벤토리모델에 의해 결정되는 레벨에 맞추어 유지하는 것이 총 소요비용을 절감할 수 있음을 보인다.

ABSTRACT

Recently, many research efforts have paid attention to spectrum pooling mechanism that provides efficient way to manage transient spectrum requests of secondary users in cognitive radio networks. Spectrum pooling is maintained by WSP (wireless service provider). WSP leases the spectrums stored in her spectrum pool to secondary users with being paid for it. It is natural that WSP tries to get profits as much as possible, which implies the WSP tries to minimize the cost required for maintaining her spectrum pool. In this paper, we model the spectrum pool into a probabilistic inventory model. Assuming secondary users' spectrum requests follow normal distribution, we give a strategic way that minimizes the maintenance cost of the spectrum pool. By a series of simulations, we show that WSP can reduce the total maintenance cost through our inventory model-based spectrum pooling than maintaining fixed inventory level.

키워드

Cognitive Radio, Spectrum Pooling, Inventory Model, Wireless Service Provider

1. 서 론

제한된 주파수 자원의 사용률을 높이고, 보다 유연한 통신 환경을 구현하기 위해 인지무선

(cognitive radio) 기술과 관련한 연구가 최근 10년 이상 이어지고 있다. 여러 연구결과 가운데, 스펙트럼풀링 (spectrum pooling) 방법은 복잡한 스펙트럼탐지 (spectrum sensing) 기능을 갖지

않은 2차사용자 (secondary user) 들이 있는 인지무선환경에 적합하다 [1, 4].

스펙트럼폴링은 1차사용자들이 사용하지 않는 주파수 대역 (CAB (coordinated access band) 이라 불리움) 을 수집 및 관리하고, 요구가 있을 때, 이를 2차사용자들에게 대여하여 사용할 수 있도록 해주는 것이다.

우리는 이러한 스펙트럼폴이 WSP (wireless service provide) 에 의해서 관리되는 상황을 고려한다. 즉, WSP는 1차사용자들에게 CAB을 구매하여 2차사용자들에게 금전적인 댓가를 받고 대여를 하는 방식을 고려한다. 여기서, 2차사용자들의 요구가 불확실하게 발생하는 경우, WSP는 CAB을 구매하는 비용뿐만 아니라, 보관 및 백오더나 스톡아웃이 발생했을 때 처리비용이 발생한다.

우리는 이 논문에서 WSP가 처리해야 하는 모든 비용 (TEC (total expected cost) 라고 명명함) 을 최소화하는 스펙트럼폴링을 제안하기 위해 인벤토리모형 (inventory model) [2]을 도입한다. 인벤토리모형은 물류창고 운영자가 불확실한 상품주문에 대해 물류창고 유지에 드는 전반적인 비용을 절감할 수 있는 모델을 제공한다. 스펙트럼폴링에서의 인벤토리는 스펙트럼폴이다. 스펙트럼폴링에 적용되는 인벤토리모형은 WSP가 불확실한 2차사용자의 요구에 대응하여 스펙트럼폴 (즉, 인벤토리) 관리에 소요되는 전반적인 비용을 절감할 수 있는 모델을 제공하게 된다.

II. 시스템 모델

논문의 페이지 제약으로 인해, 자세한 시스템 모델은 기술하지 않고, 최소 비용을 이끌어내기 위한 스펙트럼폴 관리 알고리즘을 서술하기위해 필요한 모델의 주요한 부분만 기술한다.

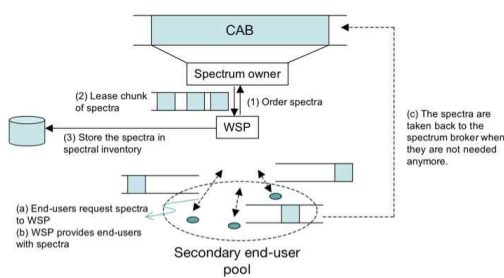


그림 1. 스펙트럼폴링 시스템 모델

우리가 제안하는 스펙트럼폴링 시스템의 모델은 그림 1에 나와있다. WSP는 스펙트럼 소유자 (1차사용자 또는 주파수할당을 관리하는 기관)로부터 주파수를 대여받아 이를 인벤토리로 관리한다 (그림 1에서 (1) ~ (3)에 해당). 엔드유저 (2차사용자)는 통신을 위해 주파수 대역이 필요할 때, WSP에게 대여를 요청하고, WSP는 이를 엔

드유저에게 대여를 한다 (그림 1에서 (a) ~ (b)에 해당). 여기서, 엔드유저는 자신이 필요로하는 만큼의 주파수 대역을 WSP에게 요구한다. 엔드유저는 대여받은 주파수 대역의 사용이 종료되면, 이를 다시 스펙트럼 소유자에게 반납한다 (그림 1에서 (c)에 해당). 즉, WSP의 역할은 스펙트럼 소유자로부터 주파수 대역을 대여하여 유지하다가, 엔드유저로부터 요청이 들어오면 유지하고 있는 주파수 대역을 재대여 하는 것이다.

여기서 WSP가 스펙트럼폴을 관리하기 위해 드는 총비용 TEC는 다음과 같이 정의 된다.

$$TEC(Q, r) = OC + SC + HC \quad (1)$$

여기서 OC (ordering cost) 는 주문하는데 소요되는 비용 (ordering cost) 이고, SC (stockout cost) 는 스톡아웃이나 백오더가 발생했을 때, 발생하는 비용이다. 그리고, HC (holding cost) 는 스펙트럼폴을 관리하는데 소요되는 비용이다.

TEC를 결정하는 변수는 한번에 주문해야하는 스펙트럼의 개수 Q 와 재주문을 해야하는 시점을 나타내는 r 이다. 만약, 재주문은 스펙트럼폴이 고갈되기 이전에 적절하게 수행되어야지, 고갈된 이후에 수행하게 되면, 새로이 주문한 스펙트럼이 스펙트럼폴로 들어오기까지 발생하는 엔드유저로부터의 요청은 모두 무시되거나 (즉, 스톡아웃으로 간주, 주문자체가 취소된다.) 백오더되게 된다. 즉, SC의 증가를 초래한다. 그리고, 재주문을 하게 될 때, Q 만큼의 스펙트럼을 주문하게 되는데, Q 가 필요이상으로 크다면, 관리해야하는 스펙트럼폴의 크기가 커져 관리비용이 추가적으로 발생하게 되고 (즉, HC의 증가를 초래) 너무 작다면 역시 스톡아웃이나 백오더로 인한 손실이 발생하게 된다. 또한, 재주문을 너무 자주하게 되면 주문비용 (OC)의 증가를 가져오게 된다. 따라서, 이 모든 것을 고려하여 Q 와 r 은 TEC를 최소화 시키는 방향으로 적절하게 선택되어야 한다.

III. 알고리즘

TEC를 최소화시키는 Q 와 r 은 다음의 알고리즘에 의해서 결정된다. 먼저, 알고리즘을 기술하는데 필요한 기호상수들의 정의는 다음과 같다.

d 엔드유저가 단위 시간동안 요구하는 스펙트럼의 평균 개수)

1) 우리는 이 논문에서 엔드유저가 d 를 직접 결정하는 것이 아니고, 엔드유저는 단지 필요 대역폭을 요구하는 것으로 가정한다. 그러면, MQAM (M-ary Quadrature Amplitude Modulation) [3] 을 가정하여, 요구대역폭을 만족시키기 위한 스펙트럼의 개수 d 가 결정될 수 있다. 본 논문에서는 이에 대한 자세한 기술은 생략한다.

α WSP가 스펙트럼 소유자에게 스펙트럼을 주문할 때, 매 주문마다 소요되는 고정 비용, 주문하는 스펙트럼의 개수와 관계없이 주문할 때마다 발생하는 비용

h 단위 스펙트럼을 WSP가 단위시간 동안 스펙트럼풀에 유지하는데 소요되는 비용

p 스톡아웃이나 백오더로 인해 발생하는 패널티 비용

$B(r)$: 매 주문 사이클 마다 발생하는 스톡아웃이나 백오더 발생 횟수의 기대값

μ : 리드타임 (lead-time)²⁾ 동안 발생하는 주문량이 정규분포를 따른다고 가정했을 때³⁾ (이 주문량의 확률분포함수를 $f(x)$ 로 정의), 그 주문량의 평균

σ : 리드타임동안 발생하는 주문량의 분산
 $G(x)$: 확률분포함수 $f(x)$ 의 보완누적분포함수

앞서 정의된 상수기호를 이용하여 알고리즘은 다음과 같이 정의된다.

Step 1: $k = 1$

Step 2: $B(r_k) = 0$ 으로 하고 다음의 등식을 계산한다.

$$Q_k = \sqrt{\frac{2d[a + pB(r_k)]}{h}} \quad (2)$$

Step 3: Q_k 를 이용하여, r_k 를 다음의 식을 이용하여 찾는다.

백오더를 하는 경우에는

$$\int_r^\infty f(x)dx = \frac{hQ}{pd} \quad (3)$$

스톡아웃인 경우에는

$$\int_{r_k}^\infty f(x)dx = \frac{hQ_k}{pd + hQ_k} \quad (4)$$

Step 4: 직전 단계에서 계산된 r_k 를 이용하여 다음을 계산한다.

$$B(r_k) = \sigma f(r_k) + (\mu - r_k)G(r_k) \quad (5)$$

Step 5: $k = k + 1$ 로 하고 Q_k 와 r_k 의 변화가 ϵ 보다 작을 때까지 Step 2부터 반복적으로 수행

2) 리드타임은 WSP가 스펙트럼 소유자에게 주문을 넣고 나서 실제 주문된 스펙트럼이 WSP에게 도착하는 데 까지 소요되는 시간을 의미한다.

3) 중심극한이론 (central limit theorem) 에 의하여 충분한 요구량이 리드타임동안 발생한다면, 이는 정규분포로 근사화 될 수 있다.

한다.

Step 3에서 r_k 를 찾는 방법은 다음과 같다. 우선 $f(x)$ 를 표준정규분포로 변환하고, 식 (3) 또는 (4)의 우변값과 가장 가까운 값을 보완누적분포함수의 통계 테이블로부터 읽는다. 그러면, 그 값에 해당하는 테이블 항목의 z 값이 r_k 가 된다. 대부분의 수확 프로그래밍 언어나 라이브러리들이 z 값을 계산해주는 함수를 제공한다. 예를 들어, GSL [5]의 함수 `gsl_cdf_gaussian_Qinv()` 함수가 있다.

IV. 시뮬레이션에 의한 성능평가

우리가 제안하는 확률인벤토리모델에 기반하는 스펙트럼풀링 기법을 평가하기 위해 수치실험을 진행한다. 실험에 사용된 주요 파라미터는 다음과 같다.

- 단위시간: 100 ticks
- 리드타임: 10 ticks
- 단위시간당 엔드유저의 주파수 평균요구량 (d): $N(1000, 250^2)$ 을 따르는 확률변수
- 주문비용 (OC): 100
- 보관비용 (HC): 4.0
- 단위 스펙트럼당 가격: 2.0
- 초기 인벤토리레벨: 200
- ϵ 0.00001

우리가 제안하는 알고리즘을 통해 최소화하는 것은 TEC, 즉 기댓값이다. 우리는 이 기댓값을 최소화하는 Q 와 r 을 결정하였을 때, 실제 총소요비용 (actual total cost) 에 어떤 영향을 미치는지 시뮬레이션을 통해 확인한다. 즉, a 와 $B(r)$ 을 위에서 언급한 $N(1000, 250^2)$ 을 따르는 확률분포로부터 샘플링하여 실제 총소요비용을 계산하여 TEC와 비교한다. 실제 총소요비용은 시뮬레이션을 1000번 수행하여 측정된 평균값을 취한다. $B(r)$ 의 평균값은 TEC를 최소화하는 과정에서 식 (5)를 통해 구해지며, 표준편차는 250이라고 가정한다.

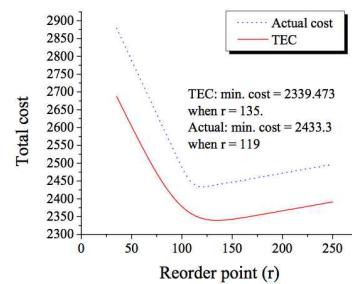


그림 2. Q 를 최적값으로 고정하고 r 을 변경시켰을 때, 측정된 TEC와 실제 총소요 비용 (actual cost) (백오더의 경우)

그림 2에 표현되듯이 r 의 변화에 따른 실제 총소요비용의 변화가 TEC의 변화와 유사함을 알 수 있다. 다만, 최소 TEC를 가져다주는 r 에서 반드시 최소 실제 총소요비용이 나오는 것은 아닌데, 이는 시뮬레이션을 더욱 많이 반복하여 더 많은 샘플의 평균을 취하면 결과적으로 TEC를 최소화시키는 r 에서 실제 총소요비용도 최소값을 가질 것으로 예상된다.

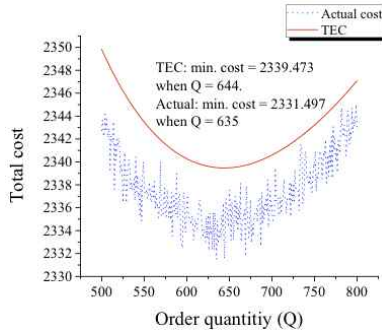


그림 3. r 을 최적값으로 고정하고 Q 를 변화시켰을 때, 측정된 TEC와 실제 총소요 비용 (actual cost) (백오더의 경우)

그림 3은 r 을 최적값으로 고정한 상태에서 Q 를 변화시켰을 때, 측정된 TEC와 실제 총소요비용을 비교한 그래프이다. 그래프에서 보여지듯이 실제 소요비용의 그래프는 큰 진동을 하고 있지만, 크게 보았을 때, 최소 TEC를 가져다주는 Q 주변에서 실제 소요비용도 최소값이 나오는 것을 알 수 있다.

시뮬레이션을 통해 보여지듯이, 항상 최소 TEC를 가져다주는 $\langle Q, r \rangle$ 이 최소 실제 총소요비용을 가져다 주지는 않는다. 하지만, 실제 총소요비용을 최소화 시켜주는 $\langle Q, r \rangle$ 값은 최소 TEC를 가져다주는 값과 큰 차이가 없음을 알 수 있다. 따라서, 만약에, 실제 총소요비용을 최소화하고자 할 때는, TEC를 최소화시키는 $\langle Q, r \rangle$ 로부터 일정하게 가까운 거리에 있는 $\langle Q, r \rangle$ 쌍들을 탐색하면 된다.

V. 결 론

스펙트럼폴링은 특정 주파수 대역을 고정적으로 할당받지 않은 엔드유저를 위해 WSP가 스펙트럼소유자로부터 스펙트럼을 대여하여 다시 엔드유저에게 재대여 하는 것이다.

확률인벤토리모형을 도입하여 스펙트럼폴링을 운영하는데 소요되는 총비용 (TEC) 를 최소화할 수 있는 방법을 제안한다. 또한, 시뮬레이션을 통해, 확률인벤토리 모델로 결정된 TEC를 최소화하는 주문량 (Q) 과 주문시점 \otimes 이 또한 실제 총

소요비용을 최소화 시키는 $\langle Q, r \rangle$ 과 크게 다르지 않음을 보인다.

이 논문에서는 충분히 많은 엔드유저를 가정하고 그들의 리드타임에 발생하는 총 스펙트럼 요구량이 정규분포를 따른다고 가정한다. 차후에, 총 스펙트럼 요구량이 정규분포가 아닌 다른 분포를 따르는 경우의 스펙트럼폴링 방법에 대해서 논하고자 한다.

참고문헌

[1] T. Berczes and A. Horvath, "A finite-source queuing model for spectrum renting in mobile cellular networks," *Prac. Elektro*, May 2014.

[2] A. Ravindran, D. T. Philips, J. J. Solberg, "Operations research: principles and practice," Second Ed., Wiley, 1987.

[3] Z. Han, Z. Ji, K. Liu, "Fair multiuser channel allocation for OFDMA networks using Nash bargaining solutions and coalitions," *IEEE Trans. Commun.* vol. 53, no. 8, pp. 1366-1376, Aug. 2005.

[4] Y. Xiao, D. Niyato, Z. Han, and K. Chen, "Secondary users entering the pool: a joint optimization framework for spectrum pooling," *IEEE J. Sel. Areas in Commun.* vol. 32, no. 3, pp. 572-588, Mar. 2014.

[5] GNU scientific library, <<http://www.gnu.org/software/gsl/>>.