논문 21-46-10-12 The Journal of Korean Institute of Communications and Information Sciences '21-10 Vol.46 No.10
https://doi.org/10.7840/kics.2021.46.10.1646

톰슨 샘플링을 적용한 강화학습 기반 Slotted ALOHA

조 영 제*, 황 경 호°

Slotted ALOHA Based on Reinforcement Learning with Thompson Sampling

Yeong-Je Jo*, Gyung-Ho Hwang°

요 약

Slotted ALOHA 무선 액세스에 톰슨 샘플링을 적용한 강화학습 기법을 제안하고 성능을 분석하였다. ϵ-greedy, Upper Confidence Bound, 톰슨 샘플링 방식의 성능을 비교하고, 톰슨 샘플링에서 파라미터 설정에 따른 throughput과 최적 성능까지 도달하는 시간의 변화를 확인하였다.

Key Words : ALOHA protocol, Reinforcement learning, Thompson Sampling, ALOHA-Q

ABSTRACT

We analyzed the performance of the reinforcement learning technique with Thompson Sampling in slotted ALOHA scheme. The performance of the E-greedy, Upper Confidence Bound, and Thompson Sampling methods were compared in simulations. The average throughput and adaptation time to reach the optimal performance in slotted ALOHA with Thompson Sampling were dependent on the parameters values.

Ⅰ. 서 론

Slotted ALOHA는 무선으로 패킷을 전송할 때 프레임을 여러 개의 슬롯으로 나눠서 패킷을 슬롯의 시작점에서만 전송한다. Pure ALOHA 대비 성능을 2배로 높여주지만, 노드들의 개수가 증가함에 따라 같은 슬롯에 패킷을 전송하여 충돌이 발생하는 비율이 높아진다. Slotted ALOHA에 강화학습을 적용하여 각각의 노드들이 프레임 내의 전송 성공률이 높은 슬롯에 패킷을 전송하여 충돌과 빈 슬롯을 줄이는 방안을 알아보고 톰슨 샘플링을 적용할 때 파라미터 설정에 따른 성능 변화를 고찰한다.

Ⅱ. 관련 연구

2.1 ALOHA-Q

ALOHA-Q는 slotted ALOHA 방식에서 노드들이 같은 슬롯에 패킷을 전송하여 충돌이 일어나거나 모든 노드가 특정 슬롯에 패킷을 전송하지 않는 현상을 보완하기 위하여 Slotted ALOHA에 Q-learning을 적용하여 충돌과 빈 슬롯을 줄여 throughput을 높이는 방식이다[1]. 각 노드들은 (1)과 같이 프레임 내의 슬롯들에 대한 Q 값을 가지고 있다. i는 해당하는 노드를 의미하고 t는 시간을 의미한다. $Q_i^t(a)$는 노드 i가 슬롯 a에 패킷을 보냈을 때마다 누적된 보상 값을 의미한다. r은 보상 값으로 노드 i가 슬롯 a를 선택하여 패킷을 보냈을 때 전송에 성공하게 되면 1을 보상받고 충돌이 발생하여 전송이 실패하면 -1을 보상으로 받게 된다. α는 학습률로 이러한 보상 결과를 얼마나 학습할지를 조절한다. 각 노드는 매 프레임에서 Q값이 가장 높은 슬롯을 선택한다.

$$Q_i^{t+1}(a) = Q_i^t(a) + \alpha(r - Q_i^t(a)) \quad (1)$$

2.2 Epsilon-greedy

ϵ-greedy 알고리즘은 탐험(Exploration)의 정도가 부족했던 greedy 알고리즘을 보완한 알고리즘으로 ϵ 값에 의하여 탐험의 정도를 조절하는 알고리즘이다. 만약 ϵ값이 20%라면 80%의 확률로 Q 값을 활용

※ 이 논문은 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No. 2018R1D1A1B07049849)
◆ First Author : (ORCID:0000-0002-0663-0746)Dept. Computer Engineering, Hanbat National University, yeongjejo@edu.hanbat.ac.kr, 학생(석사), 학생회원
° Corresponding Author : (ORCID:0000-0001-6795-8086)Dept. Computer Engineering, Hanbat National University, gabriel@hanbat.ac.kr, 정교수, 종신회원
논문번호 : 202106-146-B-LU, Received June 28, 2021; Revised July 7, 2021; Accepted July 7, 2021

(Exploitation)하여 Q 값이 가장 큰 action을 취하고, 20%의 확률로 Q 값과 상관없이 랜덤한 action을 취하여 탐험을 하게 된다.

2.3 Upper Confidence Bound (UCB)

UCB 알고리즘은 학습 초반 시도의 횟수가 작은 곳을 탐험하는 알고리즘이다. 확률적으로 랜덤하게 탐험하던 ϵ-greedy 알고리즘과 달리 신뢰 구간을 기준으로 action을 취하기 때문에 ϵ-greedy 알고리즘 대비 좋은 성능을 보여준다. UCB에서 action에 따른 값은 (2)와 같으며 UCB(a) 값이 가장 큰 action을 취한다. c는 탐험의 정도를 조절하는 값이다. t는 시간을 나타내고 $N(a)$은 t시간 동안 action a를 취한 횟수를 나타낸다[2].

$$UCB(a) = Q(a) + c\sqrt{\frac{\log(t)}{N(a)}} \tag{2}$$

2.4 톰슨 샘플링

톰슨 샘플링 알고리즘은 (3)과 같은 확률 밀도 함수를 가지는 베타 분포를 이용하고 α값과 β값을 이용하여 선택 가능 행동을 나타내는 action에 대한 각 확률분포에서 무작위로 값을 샘플링하고 그 중 보상 기댓값이 가장 높은 action을 취한다. $\Gamma()$는 감마 함수를 나타내고 θ는 0과 1 사이의 값으로 action에 대한 보상을 받을 확률을 의미하고 $1-\theta$은 보상을 받지 못할 확률을 의미한다. α는 보상받은 횟수를 나타내고 β는 보상을 받지 못한 횟수를 나타낸다. Action을 취한 후 action에 대한 보상 여부를 바탕으로 α와 β값을 업데이트하여 베타 분포의 모양을 조절한다. Action에 대해 보상을 받게 되면 α값이 증가하여 θ가 1에 가까운 쪽의 확률밀도함수 값이 커지고 보상을 못 받을 경우 β값이 증가하여 θ가 0에 가까운 쪽의 확률밀도함수 값이 커진다[3].

$$p(\theta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\theta^{\alpha-1}(1-\theta)^{\beta-1} \tag{3}$$

Ⅲ. 톰슨 샘플링을 활용한 Slotted ALOHA

모든 노드는 프레임 내의 슬롯들에 대한 베타분포 파라미터 값을 유지한다. α값은 노드가 해당하는 슬롯에 패킷을 전송할 시 성공한 횟수에 비례한 값을 나타내고 β는 패킷 전송에 실패한 횟수의 비례 값이다. 해당 슬롯에 패킷을 전송하지 않는 경우 분포는 변경되지 않는다. α와 β는 식 (4)를 사용하여 에피소드마다 갱신한다. 제안하는 방식에서는 slotted ALOHA의 한 프레임 동안의 접속이 에피소드가 된다.

$$\alpha = prior_{\alpha} + S \times K,\ \beta = prior_{\beta} + C \times K \tag{4}$$

S는 패킷 전송 성공 횟수, C는 충돌 횟수를 나타내며 K는 본 논문에서 추가로 적용하는 파라미터 값으로 성공과 실패의 가중치를 나타낸다. 처음으로 패킷을 보낼 때는 해당 슬롯의 성공률과 실패율을 모르기 때문에 α와 β의 초기 값 $prior_{\alpha}$와 $prior_{\beta}$는 각각 1로 설정하였다. 이 경우 모든 슬롯에 대해 θ의 확률밀도함수값은 1로 동일하다. 따라서 처음 패킷을 보낼 때는 랜덤한 슬롯에 패킷을 보내게 된다. 만약 노드가 패킷 전송을 성공하면 해당 슬롯의 α는 2, β는 1로 업데이트가 되고 반대로 실패한다면 α는 1, β는 2로 업데이트가 되어 베타 분포의 확률밀도함수가 변경된다. 패킷의 전송에 따른 성공과 실패에 따라 각 슬롯에 대한 α, β값이 커지면서 분포의 폭이 좁아져 샘플링 할 범위가 좁아지고 성공률이 높은 슬롯에 대한 분포가 1에 가까워지게 되어 샘플링값이 커져 전송 성공률이 높은 슬롯을 선택하여 패킷을 전송하는 확률이 높아진다.

Ⅳ. 성능 분석

강화학습 기반 slotted ALOHA에 적용하는 알고리즘들의 성능 분석을 위해 Python으로 시뮬레이션 환경을 설정하였다. UCB 알고리즘에서 c값은 0.5로 주었고 ϵ-greedy 방식의 ϵ 값은 (5)와 같이 설정하여 에피소드가 진행될수록 ϵ값이 줄어든다.

$$\epsilon = 1/((Number of episodes/100)+1) \tag{5}$$

그림 1은 슬롯의 개수를 100개로 고정하고 노드 수에 따른 평균 throughput을 나타낸다. $prior_{\alpha}$와 $prior_{\beta}$는 각각 1로 주고, 톰슨 샘플링에 적용하는 파라미터 K는 1로 설정하였다. 슬롯의 개수와 노드의 개수가 같을 때는 노드가 슬롯을 매번 랜덤하게 결정하는 Random 알고리즘을 제외한 모든 알고리즘이 1.0에 가까운 throughput을 나타내지만, ϵ-greedy 와 UCB 방식은 노드의 개수가 160개보다 커질 경우 throughput이 급격히 감소하여 노드의 개수가 슬롯의 개수보다 2배 많은 200개 일 때는 Random 알고리즘과 유사한 throughput을 보여준다. 노드의 개수가 증

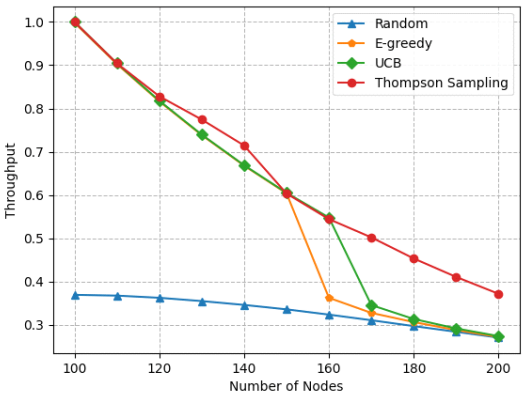

그림 1. 노드 증가에 따른 평균 처리율
Fig. 1. Average throughput vs number of nodes

가하면서 평균 throughput은 감소하지만 톰슨 샘플링 방식은 다른 방식 대비 높은 throughput을 보여주었다.

그림 2에서는 각 노드의 평균 throughput을 사용하여 Jain's fairness를 측정하였다. 1에 가까울수록 노드들의 throughput이 공평하다. Random 방식에서는 throughput은 낮지만, fairness는 1에 가까운 것을 알 수 있다. UCB나 ϵ-greedy 방식은 특정 노드들이 성공 슬롯을 계속 차지하여 fairness가 낮아지다가 노드 개수가 일정한 값을 넘어서면 노드들의 충돌이 커져서 throughput은 낮아지지만, fairness는 1에 가깝게 된다. 톰슨 샘플링 방식에서는 특정 슬롯으로 패킷을 전송하여 성공하는 노드가 유지되는 경향이 있어 노드 수가 증가할 때 fairness가 계속 낮아진다.

그림 3은 노드 수와 슬롯 개수 각각 100개인 환경에서 톰슨 샘플링 기법의 $prior_\alpha$와 $prior_\beta$ 값을 다르게 설정했을 때, 에피소드별 throughput 변화를 보여준다. $prior_\alpha$와 $prior_\beta$를 모두 1로 설정하면 UCB 알

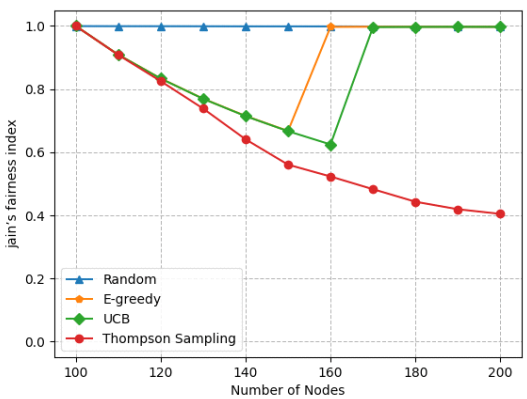

그림 2. 노드 증가에 따른 Jain's fairness index
Fig. 2. Jain's fairness index vs number of nodes

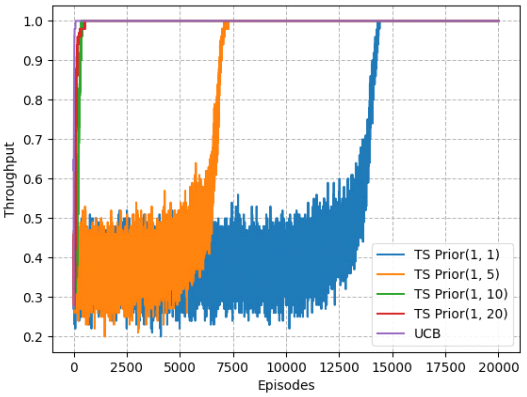

그림 3. 톰슨 샘플링 초기값에 따른 에피소드별 처리율 변화
Fig. 3. Episode throughput comparison according to prior values

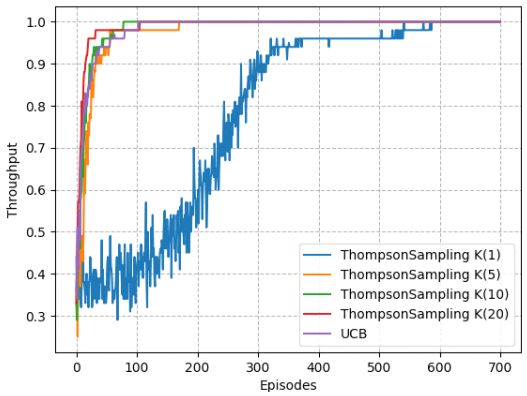

그림 4. K값에 따른 에피소드별 처리율 변화
Fig. 4. Episode throughput comparison according to K values

고리즘 대비 throughput이 1이 될 때까지 에피소드가 많이 필요하지만, $prior_\alpha$를 1로 두고 $prior_\beta$를 10 이상으로 설정하면 UCB 방식과 유사한 성능을 보여준다.

그림 4는 $prior_\alpha$와 $prior_\beta$ 를 1과 10으로 설정하고 파라미터 K를 1, 5, 10, 20으로 변경할 때의 에피소드별 처리율을 나타낸다. K가 1인 경우 대비 5이상일 때 throughput이 1이 되는데 필요한 에피소드 수가 줄어드는 것을 알 수 있다.

V. 결 론

본 논문에서는 slotted ALOHA 무선 액세스 방식에 강화학습 기반의 톰슨 샘플링을 적용하는 방안을 제안하고 시뮬레이션을 통해 성능을 확인하였다. 슬롯의 수와 노드의 수가 같거나 슬롯의 개수 대비 노드의 수가 작으면 강화학습 기반의 알고리즘들이 비슷한

성능을 보여주지만, 슬롯 수 대비 노드 수가 많아지면 톰슨 샘플링 기법이 가장 좋은 성능을 보여주었다. 하지만 톰슨 샘플링 기법에서 슬롯의 수가 많아지고, 파라미터들이 최적화가 되지 않았을 때 노드들이 최적의 성능을 달성하는데 다른 방식 대비 더 많은 에피소드가 필요한 것을 알 수 있었다. 톰슨 샘플링의 초깃값과 성공과 실패에 따른 가중치 파라미터 값에 따라 throughput이 안정화되는데 필요한 에피소드 개수가 변하는 것을 확인하였다. 추후 톰슨 샘플링 방식의 파라미터 최적화 연구를 수행할 것이다.

References

[1] R. S. Sutton and A. G. Barto, *Reinforcement Learning An Introduction*, MIT Press, 2018.

[2] C. K. Lee and S. H. Rhee, “Performance improvement of reinforcement learning based slotted ALOHA,” *J. KICS*, vol. 45, no. 11, pp. 1886-1892, Nov. 2020.

[3] D. J. Russo, B. V. Roy, A. Kazerouni, I. Osband, and Z. Wen, “A tutorial on thompson sampling,” *Foundations and Trends in Mach. Learn.*, vol. 11, no. 1, pp. 1-96, 2018.