

# ea-RED 라우터 버퍼 관리 알고리즘 성능 향상에 적합한 예측 알고리즘

임혜영<sup>o</sup> 이종현 황준

서울여대, 중앙대

naked97@sslabs.cse.cau.ac.kr, {saksi<sup>o</sup>, hjun}@swu.ac.kr

## Appropriate Forecast Algorithm for ea-RED Router Buffer Management Algorithm Performance Improvement

Hyeyoung Lim<sup>o</sup> Jun Hwang

Dept. of Computer Science and Engineering, ChungAng University, Seoul, Korea

Jonghyun Lee

Dept. of Computer Science, Seoul Women's University, Seoul, Korea

### 요 약

ea-RED(Efficient Adaptive RED)[1][2] 라우터 버퍼 관리 알고리즘 성능 향상을 위해서 ea-RED 라우터 버퍼 사이즈 변화를 예측할 수 있는 예측 알고리즘 모듈의 추가 필요성을 느꼈다. 그래서 본 논문에서는 ea-RED 라우터 버퍼 관리 알고리즘의 원형인 RED 라우터 버퍼 관리 알고리즘에 AR(AutoRegression Analysis), IIR(Infinite Impulse Response), MACD(Moving Average Convergence & Divergence), LR\_Lines(Linear Regression Lines)등의 예측 알고리즘 모듈을 적용하여 변화를 살펴보고, 결과를 비교, 분석하여 ea-RED 라우터 버퍼 관리 알고리즘 성능 향상에 가장 적합한 예측 알고리즘으로 LR\_Lines를 선정했다. ea-RED 라우터 버퍼 관리 알고리즘에 적합한 예측 알고리즘 선정을 위해서 RED 라우터 버퍼 관리 알고리즘을 대신 이용한 이유는 ea-RED 라우터 버퍼 관리 알고리즘의 경우 네트워크 상황에 따라, 버퍼 관련 파라미터 값을 수시로 바꾸기 때문에 예측 알고리즘의 정확성을 판단하는데 어려움이 있지만, RED 라우터의 경우는 버퍼 관련 파라미터 값을 변화시키지 않기 때문에, 좀 더 일관성 있고 정확한 분석을 수행할 수 있기 때문이다.

### 1. 서 론

ea-RED(Efficient Adaptive RED)[1][2]는 다수의 TCP 커넥션이 경쟁하는 병목구간에서 인터넷 라우터 버퍼를 능동적으로 관리하는 다양한 AQM(Active Queue Management) 알고리즘 중의 하나로 RED 라우터 버퍼 관리 알고리즘의 성능을 개선한 라우터 버퍼 관리 알고리즘이다. RED 라우터가 TD 라우터와 같은 네트워크 퍼포먼스를 유지하면서 TCP 커넥션 간 페어니스를 향상시키기 위해서는 link bandwidth, active 커넥션 수, congestion level 등에 대한 네트워크 상태를 고려하여 파라미터에 적절한 값을 설정해야만 한다. 문제는 다이나믹하게 변하는 네트워크 상황에 적합한 파라미터 값을 초기에 설정해주는 것이 매우 어렵다는 점이다[3][4]. ea-RED는 이런 문제를 max threshold와 min threshold 값을 네트워크 상황에 따라 동적으로 조절함으로써 해결하고자 했다. 결과적으로 기존 RED에 비해 라우터 버퍼는 50% 정도만 사용하면서도, 페어니스 인덱스(Fairness Index)[5]는 최대 41.42%까지 개선되었다.[1][2] 그러나 송신 TCP 커넥션 수의 증가에 따라 성능향상에 대한 효과가 감소되었고, 결국 기존 ea-RED 라우터 버퍼 관리 알고리즘 성능을 보다 개선할 필요성을 느끼게 되었다. 이를 위해 ea-RED 알고리즘에 예측 알고리즘 모듈을 적용하기로 결정했고, 적합한 알고리즘 선정을 위해서 RED

라우터 버퍼 관리 알고리즘에 AR(AutoRegression Analysis), IIR(Infinite Impulse Response), MACD(Moving Average Convergence & Divergence), LR\_Lines(Linear Regression Lines)등의 예측 알고리즘 모듈을 적용하여 변화를 살펴보고, 결과를 비교, 분석하여 ea-RED 라우터 버퍼 관리 알고리즘 성능 향상에 가장 적합한 예측 알고리즘을 선정할 수 있었다. ea-RED 라우터 버퍼 관리 알고리즘에 적합한 예측 알고리즘 선정을 위해서 RED 라우터 버퍼 관리 알고리즘에 예측 알고리즘을 대신 적용한 이유는 ea-RED 라우터 버퍼 관리 알고리즘의 경우 네트워크 상황에 따라, 버퍼 관련 파라미터 값을 수시로 바꾸기 때문에 예측 알고리즘의 정확성을 판단하는데 어려움이 있지만, RED 라우터의 경우는 ea-RED 라우터의 원형으로써, 버퍼 관련 파라미터 값을 변화시키지 않기 때문에 좀 더 일관성 있고 정확한 분석을 수행할 수 있었기 때문이다.

### 2. 예측 알고리즘

AR, IIR, MACD, LR\_Lines에 대한 간략한 설명을 통해 각 예측 알고리즘을 특성을 살펴볼 것이다.

#### 2.1 AR (AutoRegression Analysis)

AR알고리즘은 IIR(Infinite Impulse Response filter)과 같이 필터 디자인 산업, 그리고 물리학에서 최대 엔트로피 모델처럼 알려져 있다.

AR은 다음과 같은 식으로 정의할 수 있다.

$$x_t = \sum_{i=1}^N a_i x_{t-i} + \epsilon_t$$

그림 1. AR 정의 식

그림 1에서  $a_i$ 는 autoregression 계수를 의미하고,  $N$ 은 필터의 길이를 의미한다.  $\epsilon_t$ 는 Gaussian white noise라고 가정한다. 현재의 연속된 값들은 이전의 연속된 값들의 선형 가중치(linear weighted) 합계에 의해 평가된다. AR 알고리즘의 문제점은 연속된 주어진  $x_t$ 값을 위해 가장 적합한 값을 유추해내는 것이다.

### 2.2 IIR (Infinite Impulse Response)

IIR 필터는 무한구간 임펄스 응답 필터를 가지고 선형적인 예측을 하는 것으로 위상의 변화와 상관없이 크기변화를 위주로 예측하기 위해 사용하는 통계적인 알고리즘으로써, 값이 지나치게 확장될 때 값을 정하기 위해 일반적으로 사용된다. 과거의 출력 값을 참조하기 때문에 피드백 시스템이라고도 한다. 이 시스템은 과거의 출력 값을 이용함으로써 계산의 양이 많이 줄어들 수 있다. 동일한 계산 양으로 성능이 더 뛰어나 수 있다는 장점을 가지고 있다. 반면 계수가 잘못 선택되면 시스템이 불안정해진다 는 단점을 가지고 있다. 이러한 디지털 IIR 필터를 설계하는 대표적인 방법은 정해진 조건을 만족시키는 아날로그 필터를 디지털 필터로 변환시키는 것이다. 이것은 기존의 아날로그 IIR 필터의 경우에 기술이 많이 발달되어 있으며, 반면에 이 기술을 직접 이산시간 IIR 필터에 적용시키는 것이 어렵기 때문이다. 아날로그 필터를 디지털 필터로 변환시키는 방법에는 임펄스 불변법과 쌍일차 변환법이 있으며, 이들 방법을 통해서 아날로그 필터 디지털 필터로 변환하게 된다.

### 2.3 MACD (Moving Average Convergence & Divergence)

MACD는 주로 금융권에서 사용되는 방법으로 단기 두 이동평균 사이의 관계를 보여주는 운동량지표로서 장기 지수이동평균과 단기지수이동평균의 차이를 MACD 라 하며, 이 MACD 의 지수이동평균을 'signal' 이라 부른다. MACD 의 원리는 장기와 단기 두 이동평균선이 서로 멀어지게 되면 (diverge) 언젠가는 다시 가까워져 (converge) 어느 시점에서 서로 교차하게 된다는 성질을 이용하여, 두 개의 이동평균선이 멀어지게 되는 가장 큰 시점을 찾고자 하는 것이다. 단기간 예측 결과가 장기간 예측 결과를 넘어서는 경우 상승으로 판단하고, 장기간 예측 결과가 단기간 예측 결과를 넘어서는 경우 하향으로 판단한

다.

### 2.4 LR\_lines (Linear Regression Lines)

LR\_lines는 통계적인 툴로서 과거 값으로부터 미래의 값을 예측하기 위해 사용된다. security price의 경우에서 일반적으로 값이 지나치게 확장되었을 때 값을 결정하기 위해 사용된다. LR\_lines는 LR\_trend\_line이라고도 하는데, price와 예측선 사이에 거리를 최소화 하기 위해 price를 직선으로 최소 제곱법을 사용하여 그린 것이다. 만약 '내일 특정 가격이 어떻게 변할까?' 라고 추측할 때 오늘 가격에 상당히 근접할 것으로 생각하는 것이 논리적이다. 특히 가격이 상승추세라면 오늘 가격은 상당히 상승에 편중되어 나타날 것이다. 이와 같이 LR\_lines는 이러한 논리적인 가정의 통계학적 확증적임이라고 할 수 있다. LR\_lines의 계산 방법은 다음과 같다.

$$y = a + bx$$

$$a = \frac{\sum y - b \sum x}{n}$$

$$b = \frac{n \sum (xy) - (\sum x)(\sum y)}{n \sum x^2 - (\sum x)^2}$$

그림 2. LR\_lines 계산 식

그림 2의 계산 식에서  $y$ 는 종속변수,  $x$ 는 독립변수,  $a$ 는 절편,  $b$ 는 기울기,  $n$ 은 총 기간을 의미한다.

### 3. 시뮬레이션 분석

각 예측 알고리즘을 RED 버퍼 관리 알고리즘에 적용한 결과, AR과 IIR은 그림 3, 4와 같은 결과를 보였고, 두 개의 알고리즘 전부 값을 따라갈을 뿐 예측은 하지 못했다. MACD의 결과는 그림 5와 같다. 그림 5에서 볼 수 있듯이 MACD의 경우는 다른 알고리즘과는 전혀 다른 결과를 보여주고 있고, 역시 전혀 예측을 하지 못했다. 결론적으로 LR\_lines를 제외한 나머지 알고리즘은 다음 사이클의 평균 큐 길이 값을 전혀 예측하지 못했다. RED에 LR\_lines를 적용한 결과는 그림 6과 같다.

### 4. 결론 및 향후 과제

LR\_lines의 경우, 평균 큐 길이가 길어질 때는 비교적 정확하게 예측을 했지만, 평균 큐 길이가 짧아질 경우는 예측 값이 실제 평균 값보다 길었다. 그럼에도 불구하고 LR\_lines 예측 알고리즘이 RED의 평균 큐 길이를 예측하는데 적합하다고 생각한 이유는 RED의 특성에 기인한다. 즉, max threshold를 평균 큐 값이 침범하면 RED 버퍼관

리 알고리즘은 TD 버퍼 관리 알고리즘과 동일하게 모든 입 입력 패킷을 드롭시키는데 반해서, 평균 큐 길이가 min threshold값 보다 작아질 때는 그다지 큰 의미가 없기 때문이다. 이 실험의 결과를 통해 ea-RED 버퍼 관리 알고리즘의 평균 큐 값 변화를 예측하는데 LR lines가 적합하다는 판단을 했고, 이 예측 알고리즘을 ea-RED에 적용하여 보다 진보된 성능의 라우터 버퍼 관리 알고리즘을 구현할 수 있는 토대를 마련할 수 있었다.

### 5. 참고문헌

- [1] 임혜영, 이종현, 허의남, 황준, " NS-2를 이용한 Efficient Adaptive RED 라우터 버퍼 관리 알고리즘 성능 평가" 한국정보과학회 봄 학술발표논문집 Vol.30 No.1 2002. 4.
- [2] 이종현, 임혜영, 허의남, 황준, 김영찬, " Efficient Adaptive RED 라우터 버퍼 관리 알고리즘 디자인과 구현" 한국정보과학회 봄 학술발표논문집 Vol.30 No.1 2002. 4.
- [3] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," IEEE/ACM Transactions on Networking, vol. 1, pp. 397-413, Aug. 1993.
- [4] Go Hasegawa, Kouichi Tokuda and Masayuki Murata, "Analysis and Improvement of fairness among many TCP connections sharing Tail-Drop and RED Routers" in Proceedings of INET 2002, November 2002
- [5] Raj Jain, "Throughput fairness index: An explanation," ATM Forum Contribution 99-0045, February 1999.
- [6] LBNL, LBNL Network Simulator-ns version 1, <http://www-nrg.ee.lbl.gov/ns/>
- [7] UCB/LBNL/VINT Network Simulator - ns (version 2), <http://www-Mash.CS.Berkeley.EDU/ns/>

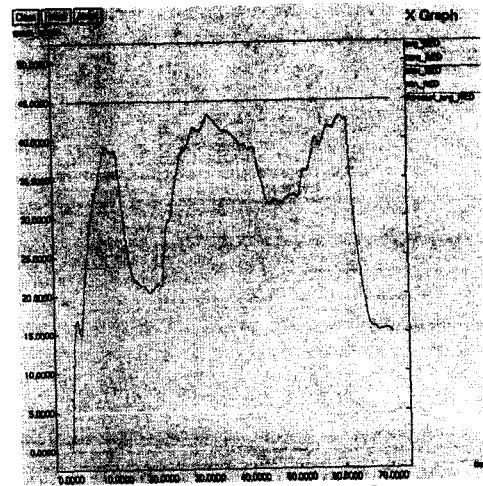


그림 4. IIR을 적용한 RED

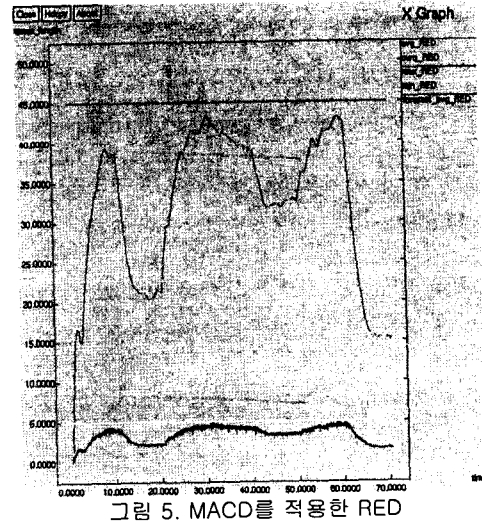


그림 5. MACD를 적용한 RED

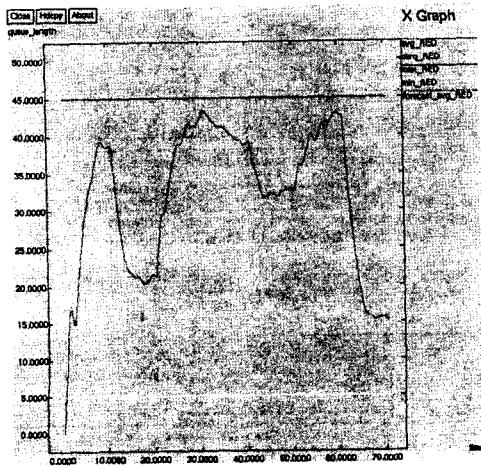


그림 3. AR을 적용한 RED

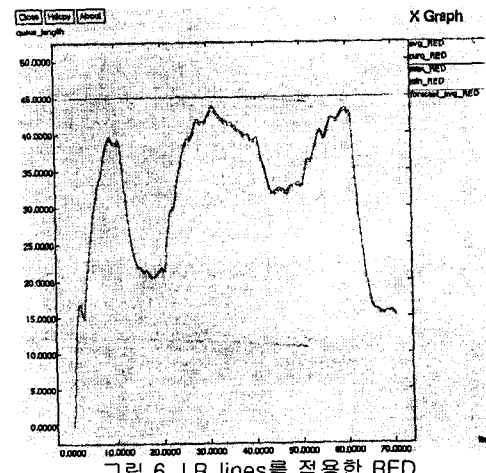


그림 6. LR\_lines를 적용한 RED