

## 고속 네트워크 환경에서 최적AQM기반의 혼잡제어를 통한 실시간 데이터 전송

황성규\*

### Real-time data transmission through congestion control based on optimal AQM in high-speed network environment

Seong-Kyu Hwang\*

\*Associate Professor, Department of Information & Communication Engg., Chosun College University of Science & Technology, Gwangju, 61453 Korea

#### 요약

TCP 통신, 패킷통신은 고품질과 고신뢰성 확보를 위해서는 전송제어 기술이 필요하다. 그러나 실시간 데이터 전송의 경우 비효율적 전송 문제가 발생한다. 이러한 문제점을 극복하여 패킷의 신뢰성 전송을 위해 일반적으로는 단순히 버퍼 수위를 지표로 하는 조기혼잡제어를 사용 하였다. 혼잡을 제어하는 시점을 버퍼 수위에 기준을 두기 때문에 혼잡제어 시점과 해지 시점의 제어가 지연된다. 따라서 본 논문에서는 버퍼 수위 지표뿐만 아니라 네트워크 혼잡을 측정하는 전송 속도와 버퍼 수위가 최적의 설정 이상 수준에 근접하도록 패킷 폐기확률을 결정하는 이상적인 버퍼 수위와 함께 분류 하였다. 제안 방법을 입증하기 위한 실험에서 설정한 이상적인 버퍼 수위에 의해 평균버퍼 수위가 유지되게 함으로 직접 제어할 수 있음을 결과로 나타내었다.

#### ABSTRACT

TCP communication and packet communication require transmission control technology to ensure high quality and high reliability. However, in the case of real-time data transmission, an inefficient transmission problem occurs. In order to overcome this problem and transmit the packet reliability, in general, early congestion control using the buffer level as an index was used. Control of the congestion control point and the cancellation point is delayed because the point at which congestion is controlled is based on the buffer level. Therefore, in this paper, not only the buffer level indicator, but also the ideal buffer level, which determines the packet discard probability, is classified so that the transmission rate and buffer level that measure network congestion are close to the level above the optimal setting. As a result, it was shown that the average buffer level can be directly controlled by maintaining the average buffer level by the ideal buffer level set in the experiment to prove the proposed method.

**키워드** : 실시간 데이터전송, 고속 전송네트워크, 혼잡제어, 최적AQM기반

**Keywords** : Real-time data transmission, High-speed transmission network, Congestion control, Optimal AQM based

Received 25 April 2021, Revised 5 May 2021, Accepted 2 June 2021

\* Corresponding Author Seong-Kyu Hwang (E-mail:okhsk@cst.ac.kr Tel:+82-62-230-8840)

Associate Professor, Department of Information & Communication Engg., Chosun College University of Science & Technology, Gwangju, 61453 Korea

Open Access <http://doi.org/10.6109/jkiice.2021.25.7.923>

print ISSN: 2234-4772 online ISSN: 2288-4165

© This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.  
Copyright © The Korea Institute of Information and Communication Engineering.

## I. 서 론

패킷 전송은 신뢰성 전송을 보장하기 위해 확인응답을 송신 단에서 수신하여 패킷의 전송 유무를 확인하는 확인응답 전송기법을 사용한다. 확인응답을 수신하기 위해 최대 수백 ms 시간지연이 발생한다. 즉 최소 왕복 지연 시간(round trip delay)이 필요하다.

전달 지연 시간(propagation delay)은 라우터에서 발생하면 최대 지연시간이 발생하여 두 배의 시간이 소요되며 이로 인한 네트워크 대역폭 효율이 나빠진다. 실시간 데이터 전송의 경우 비효율적 전송 문제가 발생한다. 라우터는 상위 계층 프로토콜로부터 전달받은 데이터를 전송하고 세션 수가 많아지면 전송하는 속도가 느려지며 데이터가 버려질 경우 TCP 혼잡(congestion)이 발생된다[1]. 혼잡이 더 증가되고 라우터와 연결된 수천개의 TCP 연결이 계속 재전송하면 혼잡이 가중되어 혼잡 붕괴(congestion collapse)가 발생된다. 이러한 TCP에서 발생하는 혼잡을 극복하기 위해서 모든 장비가 혼잡 제어 알고리즘을 실행시키는 것이 패킷의 신뢰성 전송을 위해 매우 중요하다. 지금까지는 전송지연 시간의 문제점을 해결하기 위해 TCP 혼잡 처리 방식인 RFC 2001에서 TCP Slow start, Congestion Avoidance, Fast Retransmit, Fast Recovery Algorithms의 혼잡제어 기술을 적용하고 있으며 이들 알고리즘들은 서로 상호관계를 가지고 있다[2]. Slow start와 Congestion Avoidance는 혼잡정도에 따라 크기가 바뀌는 혼잡 윈도우를 정의할 때 두 알고리즘이 독립적으로 작용하지 않고 상호작용이 된다. Fast Retransmit과 Fast Recovery은 Slow start와 Congestion Avoidance를 구현하는 방식을 수정하기 위해 만들어져 사용되어 왔다. 즉 전송지연시간의 문제점을 해결하기 위해 버퍼의 범람이 발생되면 가장 오래된 패킷을 폐기하는 Drop Head of Line 기법을 사용한다. 그런데 고속 전송 네트워크에서 효과는 거의 미미하다. 기존 조기혼잡제어는 혼잡을 제어하는 시점을 버퍼 수위에 기준을 두기 때문에 혼잡 제어 시점과 해지 시점의 제어가 지연된다. 버퍼의 수위가 제어 시점 이상으로 증가하여야 한다. 이로 인해 버퍼의 수위는 필요 이상의 패킷이 증가되며 패킷 폐기확률을 낮출 때까지 많은 시간이 필요하고 많은 패킷을 폐기하게 되며 폐기된 패킷은 재전송을 하게 된다. 이러한 기존 조기혼잡제어의 문제는 단순히 버퍼수위의 지표로 발생된다. 따라서

본 논문에서는 버퍼수위 지표뿐만 아니라 네트워크 혼잡을 측정하는 전송속도와 함께 고려되는 방법을 제안한다. 이를 위해 버퍼수위가 항상 이상적으로 유지하도록 패킷 폐기확률을 결정하게 하였다. 제안한 방법의 우수성을 입증하기 위해 무선 라우터 APR에서 예측기반 버퍼관리 기법을 적용하여 무선네트워크에서 성능을 비교 하였다. 본 논문의 구성은 다음과 같다. 2장에서는 관련연구에 대하여 설명 하고, 3장에서는 연구의 내용을 논하고, 4장에서는 성능평가를 제시한 뒤, 5장에서 결론으로 마무리 한다.

## II. 관련 연구

### 2.1. AIMD( additive increase multiplicative desaturase)

송신측 윈도우 크기를 의미하는 혼잡 윈도우 크기 cwnd(congestion window size)는 수신측의 윈도우 크기 rwnd(receiver window size)에 따라 조정이 되며 송신에서 전송한 패킷이 수신측에 전달이 되지 않을 때까지 설정된 대역폭의 전송률(cwnd)을 증가한다. 전송기법은 가산적 증가(additive increase)와 배수적 감소(multiplicative decrease) 방법이 있는데 전송 실패가 나올 때까지 왕복시간(RTT)주기에 1MSS(Maximum Segment Size)의 cwnd를 증가하고 전송이 실패할 경우 cwnd를 1/2로 줄이는 기법이다[3].

### 2.2. slow start

TCP/IP 네트워크에서 연결된 세션들의 트래픽이 전송되어 시간 내에 패킷을 처리하지 못하면 혼잡을 가중시킬 수 있다. 이러한 상황을 예방하기 위해 초기에 세그먼트의 전송 속도를 제한한다. 송수신 장비간 connection이 되면 송신측에서는 MSS의 세그먼트 하나만 송신한다. 수신측으로부터 ack가 올 때 송신할 세그먼트를 늘려서 전송할 수 있다[4]. cwnd의 초기값은 1 MSS에서 시작하며 손실이 발생하기 전까지 송신률은 지수적으로 증가하며 수신측에서 ack의 응답이 올 때 RTT(Round Trip Time)에 cwnd는 2배씩 증가한다. 송신측에서는 처음 전송시 slow start로 전송하여 송신 윈도우 size 만큼 보내거나 네트워크 혼잡을 감지할 때까지 점차적으로 크기를 늘려간다. 혼잡이 발생할 때 혼잡 회피 기능이 적용되며 그림 1의 그래프에서 손실이 발생하면 window

size를 1/2로 줄이는 기법이다.

### 2.3. Congestion Avoidance

TCP link에서는 혼잡이 발생하면 먼저 세그먼트 속도를 낮게 제어하여 전송하는 혼잡 회피 알고리즘을 가동하고 slow start를 통해 다시 속도를 높여 간다[5]. 송신측에서 ack를 받지 않은 상태에서는 데이터 양을 제한하여 송신률을 제한하는데  $LastByteSent - LastByteRead = cwnd$ 로 하며 송신측의 송신률은(bytes/sec)  $rate = cwnd/RTT$ 로 정한다.

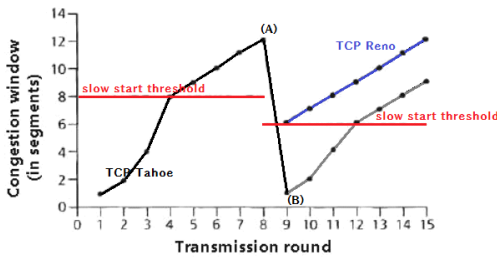


Fig. 1 Response to timeout of Congestion Avoidance

그림 1은 Congestion Avoidance의 timeout에 대한 반응을 도식화하였다. TCP Tahoe는 timeout 또는 3개의 중복 ack를 수신 상황에서는 항상 혼잡 윈도우 크기 cwnd를 1MSS로 설정하고 timeout에 의해 감지된 손실은 전혀 수신되지 않는 것으로 판단하여 심각한 혼잡 상황이라 판단한다. slow start가 발생하면 혼잡 윈도우 크기 cwnd는 1 MSS이고 혼잡 윈도우 크기 cwnd는 지수적으로 증가한다. 또한 slow start threshold에 이르면 선형적으로 증가한다 (좌표 A). 중복된 ACK가 3개 수신된 손실은 송신된 패킷이 잘 수신되었으니 일부 패킷은 전송해도 되는 것으로 판단하고 조금은 혼잡한 상황이라 판단한다. TCP Reno는 혼잡 윈도우 크기 cwnd를 1/2로 감소하고 바로 선형적으로 증가한다. 반면에 TCP Tahoe는 slow start로 시작 한다 (좌표 B). 혼잡 윈도우 크기 cwnd 값이 타임아웃 전의 1/2 크기에 도달할 때 지수적 증가는 선형적 증가로 변경한다. TCP Reno 기법이 송신 측에서 전송률이 극도로 저하되는 slow start 구간을 겪지 않아도 되므로 네트워크 전송효율을 TCP Tahoe 보다 증대시킬 수 있는 기법이다.

## III. 연구의 내용

네트워크 혼잡으로 전송 지연시간이 발생 되면 이러한 문제점을 해결하기 위해 Drop Head of Line 기법을 사용한다. 이 기법은 버퍼의 범람이 발생되면 가장 오래된 패킷을 폐기하여 문제점을 해결한다. 그러나 고속 전송 네트워크에서 이러한 기법은 효과가 거의 없다. 이를 해결하기 위해서 혼잡을 예측하여 일부의 패킷을 고의로 폐기하는 적극적인 버퍼관리 기법인 조기혼잡제어 기법(Active Queue Management)이 필요 하다.

본 논문에서는 판단 지표를 단순히 버퍼 수위뿐만 아니라 지표를 추가하여 네트워크의 혼잡 정도를 측정하여 버퍼수위가 항상 이상적으로 유지하도록 적용한다.

### 3.1. 기존 조기혼잡제어(Active Queue Management) 기법

TCP의 흐름제어 기법의 방식은 혼잡제어의 시점을 기준으로 패킷의 폐기 기능을 활성화한다. 기존 기법은 폐기와 회복이 신속히 전환되지 않고 버퍼 수위가 많은 등락의 반복이 일어나므로 지연이나 패킷 폐기가 필요 이상으로 이루어짐으로 패킷 폐기와 전송속도의 증가와 밀접한 관계가 있다.

기존 기법의 조기혼잡 예측을 통해 버퍼를 제어하는데 혼잡이 발생하는 시점을 고려하면 기존 기법은 평균 버퍼 수위가 조기 혼잡제어를 가동하기 위해 고의적 패킷 폐기가 시작되는 버퍼 수위까지 증가하여야 한다. 그러나 버퍼 수위까지 증가보다 전송속도의 증가가 우선 발생하며 평균 버퍼 수위 조기 혼잡제어의 가동은 일정 시간의 경과가 진행되어야 한다. 이러한 이유로 기존 기법은 혼잡제어 가동 시까지 시간 경과로 버퍼 수위가 필요 이상으로 발생한다. 기존 기법으로 네트워크 혼잡이 발생되면 패킷폐기 확률은 높아지고 버퍼수위가 과도하게 올라가는 상황이 발생된다.

### 3.2. 기존 조기혼잡제어 기법 종류

#### 3.2.1. Early Random Drop(ERD) 버퍼관리

ERD는 임계 수위를 설정하여 범위를 오버했을 때 혼잡을 감지하여 입력된 패킷을 폐기하고 버퍼가 오버하는 경우를 조기에 방지한다. ERD는 혼잡 발생에서 흐름제어가 되는 구간까지 시차를 줄일 수 있어 대역폭을 효율적으로 운영할 수 있다[6-7].

### 3.2.2. Random Early Detection(RED)버퍼관리

RED는 현재 버퍼 수위를 기준으로 혼잡 정도를 판단 하지만 RED는 평균버퍼수위(average queue length)를 지표로 사용한다. 실제 버퍼가 포화상태가 되기 전에 평균버퍼 수위 지표를 가지고 혼잡을 조기에 파악하고 전송속도를 제어한다. RED 방식은 여러 AQM 기법 가운데 보편적으로 널리 사용되는 방식이다[7-8].

### 3.3. 기존 조기혼잡제어 기법의 문제점

기존 조기혼잡제어는 혼잡을 제어하는 시점을 버퍼 수위에 기준을 두기 때문에 혼잡 제어 시점과 해지 시점의 제어가 지연된다. 이로 인해 버퍼의 수위가 제어 시점 이상으로 증가하여야 한다[9]. 이러한 이유로 기존 조기혼잡제어 기법의 버퍼 수위는 필요 이상의 패킷이 증가한다. 패킷 폐기확률을 낮출 때까지 많은 시간이 필요하며 많은 패킷을 폐기하게 되고 폐기된 패킷은 재전송을 하게 된다[10]. 기존 조기혼잡제어의 문제는 단순히 버퍼수위의 지표로 사용했기 때문에 발생된다. 그러므로 버퍼수위 지표뿐만 아니라 네트워크 혼잡을 측정하는 전송속도의 지표도 함께 고려되어야 한다.

### 3.4. 제안 조기혼잡제어(Active Queue Management) 기법

제안 조기혼잡제어 기법의 원리는 버퍼수위가 항상 이상적으로 유지하도록 패킷 폐기확률을 결정하는 것이다.

대역폭이 B인 라우터에서  $\tau_0$  시점에 버퍼수위가  $b\tau$  이라 하면 세션의 전송속도는  $R\tau$  이고  $\tau \geq \tau_0$  이라 가정하면 예상 버퍼수위  $Ex$ 을 식 (1)과 같이 나타낼 수 있다.

$$Ex = (R\tau - B)(\tau - \tau_0) + b\tau \quad (1)$$

라우터로 수신된 패킷들이 이상적 버퍼수위 이상일 때는 정확히  $e_m$ 의 비율로 버퍼에서 폐기하고 라우터에 수신되는 속도는 평균  $e_m R\tau_0$ 의 감소가 된다면 다음 식 (2)과 같이 얻을 수 있다.

$$Ex = b\tau + (1 - e_m R\tau_0)(R\tau - B)(\tau - \tau_0) \quad (2)$$

식(2)의  $b\tau$  예측변화율을  $-B$ 와  $R\tau - B$  사이에서 조절이 가능하며 n번째 패킷 도착 직후에 추정된 전체 TCP세션의 전송속도의 결정과 확률적으로 패킷을 폐기하는 결정 방법을 정의할 수 있다.

### 3.4.1. 전송속도 결정

제안 AQM 방식은 window 기반의 moving-average 추정방식을 사용하는데 패킷 사이즈와 시간간격의 평균값으로 전송속도를 추정한다. 장점은 전송속도 추정이 직관적이며 간편하나 계산량이 많아 큰 메모리 공간을 요구하는 단점이 있다.

본 논문에서는 auto-regressive 전송속도 추정 방식을 적용한다. auto-regressive방식은 수신된 패킷들의 크기와 시간 간격을 지수함수적 평균을 취하므로 필요한 연산을 줄일 수 있다. 패킷이 수신된 시점의 추정된 평균 패킷 크기 n번째를  $S_n$ 라하고, n번째 시간 간격으로 도착하면  $T_n$ 이라 하면 n번째 패킷이 수신된 시점의 추정 TCP의 총 전송속도  $R_n$ 는 일반적으로 다음의 식(3)과 같은 식으로 표현할 수 있다.

$$R_n = \frac{(1-\alpha) \sum_{m=0}^{\infty} \alpha^m S_{n-m}}{(1-\alpha) \sum_{m=0}^{\infty} \alpha^m T_{n-m}} \quad (3)$$

$\alpha$  제어상수는 전송 속도 추정방식의 민감도를 나타낸다.  $\alpha$  값이 0에 근사하면 바로 수신된 패킷들에 더 많은 영향을 받게 되어 트래픽 전송상황의 변화를 바로 적용할 수 있다.

### 3.4.2. 이상적인 버퍼수위 ( $Q_e$ )

라우터에서 이상적인 버퍼수위를 유지하기 위해서 패킷이 라우터에 수신된 시점을  $\tau_k$ 라하고,  $T_l$ 만큼의 시간이 경과한 지점인  $\tau_k + T_l$ 의 버퍼 수위가 최적의 설정 이상 수준인  $Q_e$ 에 근접하도록 패킷 폐기확률을 결정하여  $Q_e$ 를 정리하면 식 (4)과 같이 나타낼 수 있다.

$$Q_e = T_l(1 - P_{sc})(R_n - B) + b_i \quad (4)$$

버퍼수위가 이상적인  $Q_e$ 의 이하 수준이더라도 전송속도가 증가하면  $R_n$ 가 생성이 되어 패킷 폐기가 시작된다. 다른 전송이 시작됨으로 급격한 전송속도가 증가되고 혼잡이 발생 되더라도 패킷폐기 확률이 증가하거나 버퍼 수위가 과도하게 발생하는 것을 방지할 수 있다.

3.5. 수신측에서 도착률 측정과 패킷 유실을 측정

3.5.1. 도착률 측정[11]

패킷이 수신측에 도착한 패킷의 도착률(Throughput)을 식(5)로 나타낼 수 있다.

$$T_{tcp} = \frac{S}{t_{RTT}[\frac{\sqrt{2P}}{3} + (12\frac{\sqrt{3P}}{8})p(1 + 32P^2)]} \quad (5)$$

식(5)에서 p는 안정화된 상태의 패킷 유실율,  $t_{RTT}$ 는 RTT, S는 패킷의 크기이다. 도착률을 넘지 않도록 혼잡제어를 수행한다면 TCP 연결과의 형평성 문제를 해결할 수 있다.

3.5.2. 패킷 유실율 측정[11-12]

패킷 유실율은 한 번의 RTT(Round Trip Time)동안 잃어버린 패킷의 양의 비율을 표시한다.

$$L_{avg} = \frac{\sum_{i=0}^{m-1} W_i \cdot I_{k-i}}{\sum_{i=0}^{m-1} W_i} \quad (6)$$

$I_{k-i}$ 는 패킷 유실 간격(loss interval)이며 잃어버린 패킷 사이에 도착한 패킷의 개수를 의미한다.  $W_i$ 는 패킷 유실 간격에 대한 가중치이다. 가중치를 통해서 m개의 패킷 유실 간격에 대한 평균값( $L_{avg}$ )에 대해서 최근값에 가중치를 부여할 수 있다.

IV. 성능 평가

4.1. OPNET network simulator에 적용

제안 AQM기법과 기존 방식인 RED기법을 성능 비교를 위한 전체 구성도는 그림 2와 같다. 먼저 무선 라우터 APR(Access Point Router)에서 예측기반 버퍼관리 기법이 적용되었을 때를 전제 조건으로 한다.

그림 2는 무선라우터 APR을 통해 stations 20개가 서버에 세션을 이루며 station 전송속도는 100Mbps 대역폭으로 설정하였으며 서버와 라우터 구간의 대역폭은 100Mbps로 설정하였다. delay time은 랜덤하게 20ms에서 100ms로 설정하고 서버에서 4GB 실시간 영상을 8.8Mbps로 랜더링하여 20개의 stations에 전송을 한다.

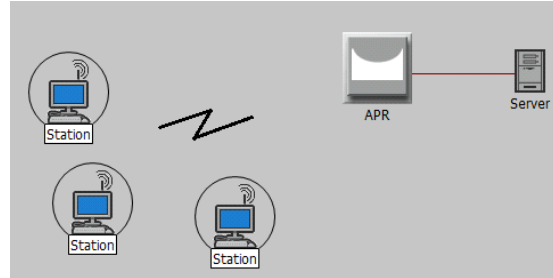


Fig. 2 Wireless network environment

병목구간인 APR의 버퍼크기는 1Mbyte로 설정하였다. RED방식과 제안AQM방식을 60분 동안 버퍼수위 변화를 모니터링 하여 평균버퍼 수위를 산출하여 명시했다.

Table. 1 Change of buffer level in comparison technique

Comparative technique	parameter	Average buffer level
RED	$buffer_{min}$ : 200KByte $buffer_{max}$ : 400KByte	470KByte
propose-AQM	$Q_e$ : 400KByte	392KByte

RED 방식의 변수 중 일정 수준의 버퍼 수준을  $buffer_{min}$ 와  $buffer_{max}$ 로 추가하면서 버퍼 수위 변화를 측정하였고 제안AQM 방식은  $Q_e$ 로 설정값을 입력하여 동일 수준의 평균버퍼 수위를 형성하도록 하였다.

표1의 성능평가에서 제안AQM 방식 파라미터의 버퍼수위  $Q_e$  근처에서 유지되었으며 제어상수  $Q_e$ 에 의해 평균버퍼 수위를 직접 제어할 수 있음을 확인하였다. RED 방식은  $buffer_{min}$ 와  $buffer_{max}$ 을 통해 직접 제어를 못 하고 평균치로 제어할 수 있음을 확인 하였다. RED 방식이 제안 AQM 방식보다 평균버퍼 수위를 직접적으로 제어할 수 있는 장점을 확인할 수 있었으며 실시간 전송의 경우 서비스 지연시간 안에서 평균버퍼 수위를 최대로 함으로 지연시간과 전송효율의 최적 파라미터를 찾을 때 제안 AQM 방식이 적용하기 유리함을 알 수 있었다.

표 1의 버퍼수위 변화를 그림3으로 도시화 했으며 빨간색 그래프는 RED기법의 변화이고 파란색 그래프는 제안AQM 기법의 변화인데 파란색 그래프가 목표 설정값의 버퍼수위  $Q_e$  근처에 버퍼수위를 유지함을 확인할 수 있다. 등락폭의 안정된 감소는 제안AQM의 개선 혼잡예측의 효과이며 버퍼수위의 변동폭이 안정적으로 감소

함으로 버퍼를 통한 지연시간의 지터가 개선될 수 있다.

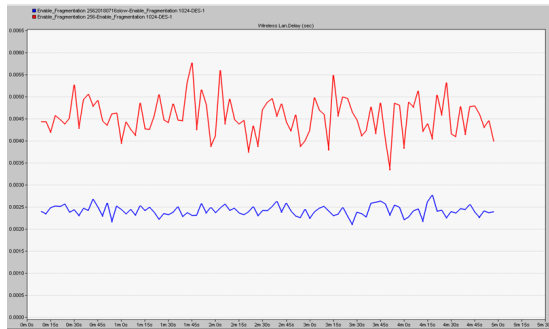


Fig. 3 Change of buffer level in comparison technique

Table. 2 Average delay time, Throughput, Packet Loss

Comparative technique	parameter	Average delay time(sec)	Delay time Standard Deviation	Throughput	Packet Loss
RED	$buffer_{min}$ : 200KByte $buffer_{max}$ : 400KByte	45.56E-02	5.0E-02	0.91	0.08
propose-AQM	$Q_c$ : 400KByte	32.64E-02	1.6E-02	0.95	0.03

표2는 평균 지연시간과 지연시간의 표준편차와 패킷의 도착률과 패킷 유실율의 측정 값이다. 제안AQM이 평균지연 시간을 유지하면서 지연시간의 표준편차가 훨씬 작은 것을 확인하였다. 누적 히스토그램으로 나타내면 RED방식은 평균값을 기준으로 넓게 분포되고 제안AQM방식은 평균지연 시간을 기준으로 급경사를 형성하면서 좁은 구간에 분포된다. 이러한 특징으로 미디어 데이터나 실시간 데이터에서는 특히 지연시간의 지터가 적은 제안AQM방식이 안정적인 운용이 가능함을 실험을 통해 확인하였다. 두 결과를 비교하면 제안AQM방식이 지연시간 지터를 줄일 수 있어 실시간 데이터에 적합함을 확인하였다.

## V. 결론

본 논문에서는 TCP 실시간 고속전송 네트워크에서 혼잡제어를 단순히 버퍼수위 지표뿐만 아니라 네트워크 혼잡을 측정하는 전송속도의 지표와 함께 고려하는

제안AQM 혼잡제어기법을 제안하였다. 이를 위해 수신된 패킷들의 크기와 시간 간격을 지수함수적 평균을 취하므로 필요한 연산을 줄일 수 있는 전송속도 추정과 버퍼 수위가 최적의 설정 이상 수준에 근접하도록 패킷 폐기 확률을 결정하는 이상적인 버퍼수위 파라미터로 추가하였다. 제안한 방법을 입증하기 위해 무선네트워크 환경에 적용하여 목표로 하는 이상적인 버퍼수위  $Q_c$  근처에서 유지되었으며 설정한 이상적인 버퍼수위  $Q_c$  2700Byte에 의해 평균버퍼 수위 2600Byte에 근접하게 유지되고 있어 버퍼수위를 제어할 수 있음을 결과로 나타내었다.

제안 조기혼잡제어 기법이 특히 혼잡 지표를 혼잡 발생 시점과 제어 시점과의 지연시간이 발생하였는데 예상 전송속도와 버퍼 수위를 파라미터로 적용하여 평균 지연 시간을 유지하면서 지연시간의 표준편차가 훨씬 작은 것을 확인하였다. 따라서 본 논문에서 제안한 방법은 패킷들의 크기와 시간 간격을 지수함수적 평균을 적용함으로써 연산의 복잡도를 줄일 수 있어 지연시간 지터를 줄일 수 있으므로 실시간 데이터에 전송에 적용될 수 있을 것으로 판단된다.

## REFERENCES

- [1] C. Raiciu, M. Handley, and D. Wischik, "Coupled multipath-aware congestion control," *IETF Internet Draft*, Jul. 2010.
- [2] D. Wischik, C. Raiciu, and M. Handley, "Balancing Resource Pooling and Equipoise in Multipath Transport," *ACM SIGCOMM*, 2010.
- [3] D. Wischik, C. Raiciu, A. Greenhalgh, and M. Handley, "Design, implementation and evaluation of congestion control for multipath TCP," *Proc. of the USENIX NSDI'11 Conf.*, Mar. 2011.
- [4] A. Ford, C. Raiciu, M. Handley, and O. Bonaventure, "TCP Extensions for Multipath Operation With Multiple Address," *Internet Requests for Comments, IETF, RFC 6824*, Jan. 2018.
- [5] C. Raiciu, M. Handley, and D. Wischik, "Coupled Congestion Control for Multipath Transport Protocols," *Internet Requests for Comments, IETF, RFC 6356*, Oct. 2018.
- [6] Y. Z. Ben, D. K. John, and Anthony, "Tapestry: An infrastructure for fault-tolerant wide-area location and routing," University of California, Berkeley: CA, Technical

- Report CSD-01-1141, 2001.
- [ 7 ] R. Khalili, N. Gast, M. Popovic, U. Upadhyay, and J. Y. LeBoudec, "Non-Pareto Optimality of MPTCP: Performance Issues and a Possible Solution," *IEEE/ACM Transactions on Networking*, vol. 21, no. 5, pp. 1651-1665, 2017.
- [ 8 ] C. Raiciu, C. Paasch, S. Barre, A. Ford, M. Honda, F. Duchene, O. Bonaventure, and M. Handley, "How hard can it be? Designing and implementing a deployable multipath tcp," in *Proceeding of 9<sup>th</sup> USENIX Symposium on Networked Systems Design and Implementation, Boston*, pp. 2-9, Apr. 2017.
- [ 9 ] M. Carbone and L. Rizzo, "Dumynet Revisited," *ACM SIGCOMM Computer Communication Review*, vol. 40, no. 2, pp. 12-20, 2010.
- [10] J. H. Hwang and J. Yoo, "Packet scheduling for Multipath TCP," in *Proceeding of the 7<sup>th</sup> International Conference on Ubiquitous and Future Networks*, Sapporo, pp. 177-179, Jul. 2015.
- [11] D. H. Youm, Ravi Yadav, "Load Balancing Strategy using Round Robin Algorithm", *Asia-pacific Journal of Convergent Research Interchange*, HSST, ISSN : 2508-9080, Vol.2, No.3, pp. 1-10, Sep. 2016.  
<http://dx.doi.org/10.21742/APJCRI.2016.09.01>.
- [12] N. Sharoon. "Dynamic Path construction in Multi-Hop Wireless Networks," *Asia-pacific Journal of Convergent Research Interchange*, HSST, vol. 2, no. 2, pp. 19-25, Jun. 2016. <http://dx.doi.org/10.21742/APJCRI.2016.06.03>.



**황성규 (Seong-Kyu Hwang)**

2007년 전주대학교 컴퓨터교육학과 (교육학석사)

2015년 조선대학교 정보통신공학과 (공학박사)

2017년~ 현재 조선이공대학교 ICT융합과 조교수

※관심분야: 통신 프로토콜, 네트워크 설계, 통신 네트워크