

네트워크 혼잡제어를 위한 FQ-CoDel 매개변수 최적화 모델링

정 소 이*, 이 충 희*, 박 형 원*, 권 대 훈**, 김 재 현^o

Optimization and Modeling of FQ-CoDel Parameter for Network Congestion Control

So-Yi Jung*, Choong-Hee Lee*, Hyung-Won Park*, Dae-Hoon Kwon**, Jae-Hyun Kim^o

요 약

전술 통신망은 작전 운용상 정보의 중요도에 따라 군사 우선순위를 달성하기 위한 목적으로 응용 서비스마다 플로우를 분류할 필요가 있다. 플로우 기반의 서비스는 트래픽의 속성에 따라 네트워크 혼잡 상황이 달라지고 상이한 큐잉 지연과 패킷 손실을 초래한다. 본 논문에서는 개별 응용 서비스의 QoS(Quality of Service)를 보장하고 네트워크 혼잡제어를 관리하기 위하여 큐 관리 및 스케줄러 기법이 결합된 FQ-CoDel(Flow Queue-Controlled Delay) 매개변수를 적응적으로 최적화하는 알고리즘을 설계한다. 제안하는 큐 관리 기법은 적절한 시점에 패킷을 폐기함에 따라 큐잉 지연을 감소시키고 큐 안정화 시점을 앞당겨 준다. 제안하는 스케줄링 방식은 목적에 따라 개별 플로우를 우선처리하거나 전체 플로우를 공평처리하는데 장점이 있다.

Key Words : AQM, Controlled delay, Flow Queue-CoDel, Target delay, Congestion control

ABSTRACT

A tactical network needs to categorize flow by every application service with the aim of achieving military priorities depending on the importance of information in operational employment. The flow-based services make different network congestion control depending on traffic, and it causes to happen to different queuing delays and packet loss. In this paper, we proposed an algorithm to dynamically optimize the parameter of flow queue-controlled delay(FQ-CoDel), which combines queue management and scheduler, to ensure the quality of service(QoS) of individual application services and to manage network congestion control. The proposed method of queue management reduces queuing delays and advances the timing of queue stabilization as packets are dropped at an appropriate time. The proposed method of scheduling has the advantage of prioritizing particular flow or fairing the entire flow according to the purpose of the algorithm.

※ 본 연구는 국방과학연구소가 지원하는 미래 전술망의 통합네트워크 구조 및 상호연동 Gateway 기술 연구 과제의 일환으로 수행되었습니다. (UE161127ED)

• First Author : Ajou University Department of Electrical and Computer Engineering, sogloomy@ajou.ac.kr, 학생회원

^o Corresponding Author : Ajou University Department of Electrical and Computer Engineering, jkim@ajou.ac.kr, 종신회원

* LIG NEX1, Korea, choonghee.lee@lignex1.com, 정희원; hyungwon.park@lignex1.com, 정희원

** Agency for Defense Development, Korea, dhkwon@add.re.kr

논문번호 : 201910-237-C-RU, Received October 21, 2019; Revised November 8, 2019; Accepted November 12, 2019

I. 서론

실시간 멀티미디어 데이터는 손실에 민감하며 네트워크 상황에 따라 서비스 품질이 결정된다. 현재 네트워크 시스템은 사용자의 트래픽이 증가함에 따라 혼잡 상황이 빈번하게 발생한다. 네트워크 혼잡 상황은 큐잉 지연과 패킷 손실을 초래하며, 제한된 시간 내에 서비스를 전송하지 못하게 되고 사용자의 QoS(Quality of Service) 보장에 영향을 준다. 현재 라우터로 유입되는 패킷들은 전통적으로 종단 탈락(tail drop) 방식으로 처리된다. 종단 탈락 방식은 큐에 저장 공간이 남아있을 때까지 가능한 한 많은 패킷을 저장하고, 큐에 저장 공간이 없을 때 나중에 도착하는 패킷을 무조건 탈락시킨다. 큐가 가득 차는 상황이 자주 발생하면 네트워크 혼잡도가 증가하고, 모든 TCP 접속이 동시에 느려지게 된다. 또한 트래픽 흐름 속에서 버퍼 공간을 불공정하게 배분하고, 네트워크 이용 효율 측면에서 성능이 저하된다. 이러한 문제점을 해결하기 위해 AQM(Adaptive Queue Management)의 필요성이 대두되었다¹⁾. AQM은 큐 내부의 양을 관찰하여 혼잡을 예측하고 탐지하는 알고리즘이며, 수신단의 라우터에서 사용되고 있다. 큐 내부의 양이 임계값을 초과하면 혼잡이 발생하였다고 예측하고 라우터로 유입되는 패킷을 폐기하거나 미리 혼잡 상황을 알려 전송 노드에서 전송률을 줄여 혼잡을 피하도록 한다. 대표적으로 큐에 데이터가 차면 패킷을 차례대로 폐기하는 FIFO(First In First Out) 방식과 큐가 다 채워지기 전에 라우터에서 큐에 인입되는 패킷을 확률적으로 폐기하는 RED(Random Early Detection) 방식이 있다.

기존 AQM 알고리즘이 평균 큐 사이즈 기반으로 동작하는 것과 달리 패킷이 실제 큐에 머무르는 시간을 측정하여 패킷을 관리하는 CoDel(Controlled Delay) 연구가 새롭게 제안되고 있다. CoDel 프로토콜은 패킷이 큐에 체류하는 지연시간(sojourn time)을 계산하여 네트워크의 혼잡을 예상한다. 계산된 지연시간이 기준을 넘는 경우 큐의 헤드부터 패킷을 폐기한다. CoDel 알고리즘은 큐 크기, 큐 크기의 임계값, 링크 사용률, 큐 점유 시간, 왕복 지연시간과 같은 매개변수에 독립적이기 때문에 다양한 환경에서 구성하기 용이하고, 통신망에서 패킷 전송 시간이 버퍼로 인해 예상보다 오래 지연되는 현상인 버퍼블로트를 다루는데 가장 적합하다고 여겨진다²⁻⁴⁾. FQ(Flow Queue)-CoDel 프로토콜은 사용자의 QoS 요구사항을 반영하기 위해 플로우를 관리하기 위한 방식으로 CoDel 프

로토콜의 큐 내부 삭제 정책에 DWRR(Deficit Weighted Round Robin) 스케줄링 기법을 결합한 방식이다. FQ-CoDel 알고리즘은 매 플로우마다 서브 큐를 생성하고, 각 서브 큐에 기존 CoDel 알고리즘을 적용하여, QoS 특성에 맞는 품질 제어 서비스에 적합하다⁵⁻⁶⁾.

전술 통신망은 군 운용체계에 따라 보장형/비보장형, 음성/비디오, 파일/단문메시지 전송 등 다양한 응용 서비스가 제공되고 있다. 또한 작전 운용상 정보의 중요도에 따라 군사 우선순위를 달성하기 위한 목적으로 응용 서비스마다 플로우를 분류할 필요가 있다. 일반적으로 응용 서비스마다 발생하는 트래픽의 속성이 다르기 때문에 데이터 속도, 지연, 신뢰성 등 상이한 네트워크 요구사항을 충족하기 위해서는 개별 큐를 동적으로 관리하는 기법이 필요하다. 본 연구의 주된 목적은 다양한 응용 서비스가 존재하는 전술 환경에서 서비스의 QoS 보장하고 네트워크 혼잡제어를 위하여 FQ-CoDel 매개변수를 적용적으로 최적화하는 알고리즘을 제안하고 그 성능을 분석한다. 본 논문은 다음의 순서로 구성된다. 2장에서는 기존 FQ-CoDel 매개변수 설정 방법을 분석하고, 3장에서는 매개변수 최적화를 제안한다. 4장에서는 시뮬레이션 성능을 분석하고 5장에서 결론을 맺는다.

II. 지연시간 기반 AQM 기법의 매개변수 설정

CoDel 알고리즘은 지연시간을 기반으로 동작하기 때문에 허용 대기 지연시간인 target delay 값을 이용하여 큐 내부 삭제 정책을 결정하고, interval 값을 이용하여 정책이 업데이트되는 빈도를 결정한다. FQ-CoDel 알고리즘에서는 매 라운드 상이한 플로우에서 처리되는 패킷의 양인 quantum 값을 이용하여 플로우 별 가중치를 결정한다. CoDel과 FQ-CoDel 알고리즘을 정의한 RFC 문서^{2,5)}에서는 일반적인 네트워크 상황을 가정하고 target delay, interval, quantum의 기준이 되는 값을 다음과 같이 설정하였다.

- target delay : 5 ms
- interval : 초기 값 100 ms, 다음 폐기 타이밍은 패킷 폐기 횟수의 제곱근에 반비례하여 차례로 감소
- quantum : 고정 값 1,514 bytes, 이더넷 MTU(1,500 bytes) + 헤더(14 bytes) 크기로 설정

실제 네트워크 환경에서는 발생하는 트래픽의 양, 병목구간의 링크 용량, 왕복 지연시간 등을 고려하여

적절한 target delay와 interval 값을 설정할 필요가 있다. 이와 관련된 연구는 특정 네트워크 환경에서 매개변수를 변화시키면서 필요한 요구사항에 맞게 튜닝하는 Kulatunga의 연구¹⁷⁾와 최적화된 매개변수 값을 도출하고 네트워크 환경에 맞추어 적응적으로 변화하는 Ye의 연구¹⁸⁾로 구분된다. Kulatunga의 연구는 고정된 네트워크 환경에서 병목구간의 사용률을 높이고, 큐 지연시간을 제한하기 위한 target delay와 interval의 조합을 찾는 것이다. 네트워크 환경이 변하는 상황마다 매개변수 조합을 새롭게 찾아야하기 때문에 실제 활용도가 낮다는 단점이 있다. Ye의 연구는 TCP 유체 모델을 이용하여 왕복 지연시간, 병목구간 대역폭, TCP 플로우 수 기반의 피드백 제어를 디자인한다. 네트워크 상황에 따라 TCP goodput을 최대화하며 CoDel 시스템의 안전한 범위 내에서 지연 요구사항을 만족하는 target delay를 최적화하여 도출한다. Ye의 연구는 매개변수 중 target delay 하나만 고려하고 있으며, TCP 세션을 플로우 별로 구분하고 처리한다는 제약이 있다.

III. 제안하는 FQ-CoDel의 최적화 설계

본 절에서는 전술 트래픽에 따라 네트워크 혼잡 및 서비스 성능에 영향을 미치는 FQ-CoDel의 세 가지 매개변수를 적응적으로 최적화하는 설계 방법을 단계적으로 설계한다.

3.1 target delay 최적화

FQ-CoDel 알고리즘은 패킷이 큐에 인입되는 시점의 시간 스탬프를 기록하고 해당 패킷이 머무는 지연시간을 추적하여 목표 지연시간인 target delay 값과 비교한다. 본 연구에서는 전술 통신망에서 병목구간 대역폭 크기를 알고 있는 환경을 가정한다. 병목구간의 대역폭의 크기 C 와 목표 지연시간 t_{target} 을 이용하여 목표 지연시간을 넘지 않기 위한 시간당 허용 가능한 큐의 길이 β_{target} 을 수식 (1)과 같이 계산할 수 있다.

$$\beta_{target} = C(1 + t_{target}) \quad (1)$$

β_{target} 은 큐에서 처리 가능한 패킷의 양과 허용 지연시간을 고려하여 향후 처리 가능하다고 예상되는 패킷의 양의 합으로 계산된다. 하나의 큐에 다수의 플로우가 인입되는 FQ-CoDel 프로토콜에서 네트워크 환경에 따라 최적의 target delay 값을 결정하는 방법

은 Ye의 연구¹⁸⁾에서 다음과 같이 제안하고 있다.

$$t_{stable,s} = \sqrt[3]{16CN_s^2I_0^2(\lambda_0 - C) - CT_p} \quad (2)$$

수식 (2)는 네트워크 환경에 따라 target delay의 안정 범위의 최댓값을 나타낸다. 여기서 고려하는 네트워크 환경은 병목구간 대역폭 C , 플로우의 수 N_s , 전파 지연시간 T_p , 최소 허용 큐 길이 β_{min} 을 초과하는 최초 패킷 도착률 λ_0 가 있다. 수식 (2)를 FQ-CoDel 프로토콜에 적용하기 위해서 동일 서비스 클래스에 속한 모든 플로우는 동일한 target delay 값을 사용함을 가정하고, target delay 값은 서비스 클래스 단위로 계산한다.

$$t_s = \min\{\max\{t_{min}, t_{stable,s}\}, t_{max,s}\} \quad (3)$$

$$t_{opt} = \max\left\{t_{min}, \frac{t_s}{N_s}\right\} \quad (4)$$

수식 (3)은 동일 서비스 클래스 내 모든 플로우의 target delay의 합을 나타내고, 수식 (4)는 개별 플로우의 optimal target delay를 나타낸다. 수식에서 사용하는 파라미터는 표 1에 나타나 있으며 서비스 클래스 s 에 따라 가변하는 파라미터는 N_s 와 $t_{max,s}$ 이다. $t_{max,s}$ 는 플로우 수와 EI-CER(End Instrument-Customer Edge Router) 지연시간의 곱으로 계산된다. 개별 플로우에 적용되는 최적의 target delay는 하한값과 비교하여 최종값을 결정한다.

표 1. target delay 최적화에 사용되는 매개변수
Table 1. Parameters for target delay optimization

Parameter	Description
C	병목구간 대역폭
N_s	서비스클래스 s 에서 플로우 수
I_0	최초 interval
β_{min}	시간당 허용 가능한 최소 target 큐 길이 ($\beta_{min} = C(1 + t_{min})$)
λ_0	최초 β_{min} 을 초과하는 패킷 도착률
T_p	전파 지연시간
d_{EI-CER}	End Instrument-Customer Edge Router 지연시간
t_{min}	target delay 최솟값
$t_{max,s}$	각 서비스 클래스에서 target delay 최댓값 ($t_{max,s} = N_s \times d_{EI-CER}$)

3.2 interval rate 최적화

FQ-CoDel 알고리즘은 패킷 폐기 시점이 도래하면 허용 가능한 큐 길이가 넘는 모든 패킷을 폐기한다. 패킷을 폐기하는 간격에 따라 패킷이 폐기되는 총량과 패킷 손실률에 영향을 준다. 또한 패킷을 폐기하는 시점에 따라 큐의 혼잡도 및 안정화 시점에 영향을 준다.

$$I(t) = \sum I(n) = \sum \frac{100}{n^{0.5}} \quad (5)$$

수식 (5)는 기존 CoDel 알고리즘에서 패킷이 폐기되는 시점이다. n 은 패킷 폐기 횟수를 나타내고, $I(n)$ 은 패킷 폐기 횟수에 따른 패킷 폐기 interval을 나타낸다. 최초 패킷 폐기 시점은 100 ms 이며, 다음 패킷 폐기 시점은 패킷 폐기 횟수의 제곱근에 반비례하여 차례로 감소한다. 일반적으로 CoDel 알고리즘은 병목 구간에서 과도한 패킷이 한꺼번에 발생하여 예상 처리시간이 비 이상적으로 증가하는 현상을 예방하기 위해 사용된다. 따라서 패킷 폐기 시점이 도래하더라도 폐기할 패킷이 없다면 차례를 넘기게 되고, 반대로 과도한 트래픽이 발생하여 네트워크가 매우 혼잡한 상황이라도 패킷 폐기 시점까지 기다렸다가 패킷을 폐기해야 한다. 대역폭이 작은 병목구간일수록 기하급수적으로 패킷이 큐에 쌓이고 대기시간을 증가시키는 성능 저하를 야기한다. 따라서 인입되는 패킷의 양과 네트워크 환경에 따라 신속한 큐 안정화를 위해 적절한 시점에서 패킷을 폐기하는 알고리즘이 필요하다.

제안하는 방식은 최초 패킷 폐기 시점에서 손실되는 패킷의 양을 확인하여 향후 트래픽을 예측하고 다음 폐기 시점을 결정한다.

$$\gamma_j = \left\lceil \frac{\beta_{opt}}{p_{100,j}} \right\rceil \quad (6)$$

$p_{100,j}$ 는 플로우 j 에서 최초 패킷 폐기 시점인 100 ms에 폐기된 패킷의 수를 나타낸다. β_{opt} 는 수식 (4)에서 도출한 optimal target delay를 수식 (1)에 대입하여 계산된 허용 가능한 큐의 길이이다. 수식 (6)은 $p_{100,j}$ 에서 β_{opt} 대비 상대적인 패킷 손실 비율을 나타낸다.

$$\alpha_j = \left\lfloor \frac{\gamma_j}{10} \right\rfloor \times 0.1 \quad (7)$$

$$I_{opt}(n) = \frac{100}{n^{0.5+\alpha_j}} \quad (8)$$

수식 (7)은 γ_j 를 적절한 비율로 매핑한 optimal interval rate 지수이다. 수식 (8)은 γ_j 를 적용한 새로운 optimal interval rate이다.

3.3 quantum 최적화

FQ-CoDel 알고리즘은 서비스 플로우마다 논리적인 큐를 생성하고 큐를 개별적으로 처리하기 위해 DWRR 스케줄링 방식을 사용하고 있다. 전술 통신망에서 응용 서비스마다 상이한 QoS 요구사항을 보장하기 위해서는 플로우 별 가중치인 quantum 값을 적절하게 분배해야 한다. FQ-CoDel 프로토콜에서 quantum과 지연시간의 관계를 수식으로 정의하면 다음과 같다.

$$r_j = \left\lceil \frac{p_{avg,j}}{q_j} \right\rceil \times n_j \quad (9)$$

수식 (9)는 플로우 j 에서 모든 패킷을 처리하는데 필요한 총 지연시간을 예측한 값이다. q_j 는 quantum 값, $p_{avg,j}$ 는 평균 패킷의 크기, n_j 는 패킷의 개수이다. 모든 플로우가 RR(Round Robin) 스케줄링 방식으로 1 회 처리되는 것을 round로 정의하고, 지연시간의 단위는 round로 한다. quantum 최적화에서 사용하는 파라미터는 표 2에 정리되어 있다. 본 연구에서는 수식 (9)를 이용하여 목적에 따라 두 가지 quantum 최적화 방식을 제안한다.

$$\begin{aligned} &\text{minimize} \quad \hat{r}_j \quad (j \text{ is the best priority}) \\ &\text{subject to} \quad \sum_{j=1}^k \hat{q}_j \leq \sum_{j=1}^k q_j, \quad \hat{r}_j < r_j \\ &\text{output :} \quad q_{opt} = \hat{q}_j \end{aligned} \quad (10)$$

수식 (10)은 특정 플로우 j 를 최우선 처리하는 최적화 수식이다. 목적함수 \hat{r}_j 는 주어진 제약조건에 맞

표 2. quantum 최적화에 사용되는 매개변수
Table 2. Parameters for quantum optimization

Parameter	Description
q_j	플로우 j 에서 사용되는 default quantum
\hat{q}_j	플로우 j 에서 사용되는 optimal quantum
n_j	플로우 j 에서 처리해야 되는 패킷 개수
$p_{avg,j}$	플로우 j 에서 평균 패킷 크기
r_j	플로우 j 에서 예측되는 round
k	우선순위 k 에서 플로우 수 ($k = \sum N_j$)

는 optimal quantum \hat{q}_j 를 이용하여 모든 패킷을 처리 하는데 필요한 round이다. 한 번의 round에서 처리되는 quantum 합은 유지하면서 특정 플로우의 지연시간을 단축시킨다.

$$\begin{aligned} &\text{minimize} \quad L_1 = \sum_{j=1}^n |r_i - r_j| = \|r_i - r_j\|_1 \quad (i \neq j) \\ &\text{subject to} \quad \forall \hat{r}_j \leq \min(r_j) \\ &\text{output : } \hat{r}_j, \text{ then calculate } q_{opt} = \hat{q}_j = \frac{n_j P_{avg,j}}{\hat{r}_j} \end{aligned} \quad (11)$$

수식 (11)은 모든 플로우에서 최종적으로 마지막 패킷이 처리 완료되는 지연시간을 최소화하는 최적화 수식이다. 목적함수는 모든 플로우에서 생성될 수 있는 L1 norm을 나타낸다. 모든 \hat{r}_j 는 default quantum 값을 이용하여 예측한 round 보다 작아야 하고, round 간 편차를 최소화하는 optimal quantum 값을 찾는다.

3.4. FQ-CoDel 설계 알고리즘

기존 FQ-CoDel은 응용 서비스 및 네트워크 환경과 무관하게 고정된 파라미터를 사용한다. 반면 제안하는 FQ-CoDel은 응용 서비스마다 상이한 큐잉 지연 요구사항을 수용하기 위해 개별 서비스마다 논리적인 큐를 생성하여 파라미터를 개별적으로 적용한다. 알고리즘 1은 target delay, interval rate, quantum 매개변수 최적화를 이용한 전체 FQ-CoDel 설계 절차이다. 알고리즘은 크게 두 단계로 구분된다. 첫 번째 단계에서는 target delay를 서비스 클래스 단위로 계산하고, interval rate는 플로우 단위로 계산한다. 두 번째 단계에서는 우선순위마다 목적을 다르게 하여 플로우 단위로 quantum을 계산한다. 알고리즘 1에서 k는 우선순위, s는 서비스 클래스, j는 플로우의 개수를 나타낸다.

알고리즘 1. 적응형 FQ-CoDel 알고리즘
Algorithm 1. Adaptive FQ-CoDel algorithm

Algorithm 1. Optimize $t_{opt}, I_{opt}(n), q_{opt}$

- 1: **Procedure** OPTIMIZE $t_{opt}, I_{opt}(n), q_{opt}$
- 2: **Input:** $C, N_s, T_p, \lambda_0, P_{avg,i}$
- 3: **Step 1.**
- 4: **for** $s = 1: \max(\text{service class})$ **do**
- 5: Calculate optimal *target delay* t_{opt} by (4)
- 6: **end**
- 7: **for** $j = 1: \max(\text{dynamic flow})$ **do**
- 8: Calculate optimal *interval rate* $I_{opt}(n)$ by (8)
- 9: **end**
- 10: **Step 2.**
- 11: **for** $k = 1: \max(\text{level priority})$ **do**
- 12: **for** $j = 1: \max(\text{dynamic flow})$ **do**
- 13: Calculate optimal *quantum* q_{opt} by (10) or (11)
- 14: **end**
- 15: **end**
- 16: **end procedure**

IV. 성능 분석

본 절에서는 제안하는 FQ-CoDel 알고리즘의 성능을 검증하기 위하여 성능분석을 위한 네트워크 구조를 설계하고 환경을 정의하였다. 단계별 성능 검증을 위해 첫 번째 단계에서는 총 패킷 폐기량, 큐 안정화 시점, outdated 패킷의 양을 분석하고, 두 번째 단계에서는 round 시간을 비교한다.

4.1 FQ-CoDel 구조 및 네트워크 환경

본 연구에서 제안하는 FQ-CoDel 알고리즘의 성능 분석을 위하여 시스템 구조는 그림 1, 표 3과 같이 2-level priority, 5-service class, 64-dynamic flow로 설계한다. 각 전술 우선순위의 차등 처리를 위해 1순위(Highest priority)는 보장형 응용 서비스(Assured Voice, Assured Multimedia Conferencing)를 최우선

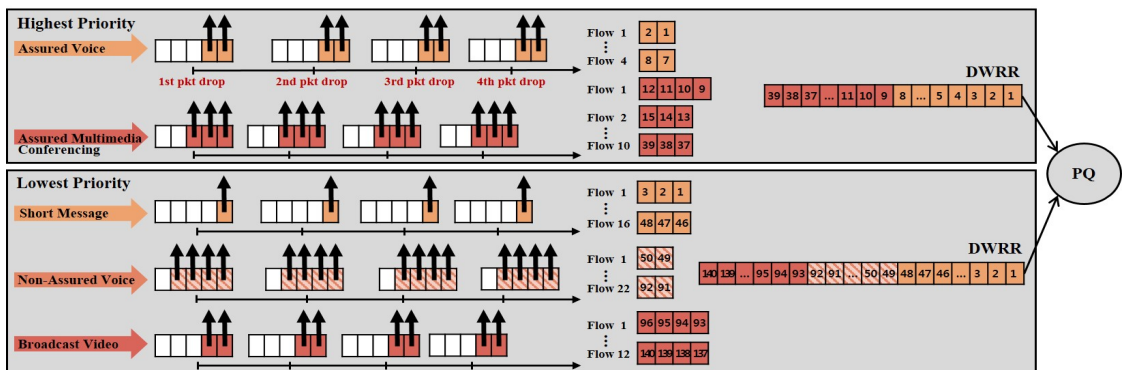


그림 1. FQ-CoDel 네트워크 구조
Fig. 1. A network structure of total FQ-CoDel

표 3. 성능분석을 위한 FQ-CoDel 구조
Table 3. Structure of FQ-CoDel for performance analysis

Application service		Number of flow
Highest priority (1순위)	Assured Voice	4
	Assured Multimedia Conferencing	10
Lowest priority (2순위)	Short Message	16
	Non-Assured Voice	22
	Broadcast Video	12

표 4. 성능분석 환경
Table 4. Performance analysis environment

Parameter	Value
병목구간 크기(B)	24 Mbps
병목구간 대비 트래픽 양 (α)	Assured Voice = 1.4 Assured Multimedia Conferencing = 1.7 Short Message = 1.3 Non-Assured Voice = 1.5 Broadcast Video = 1.8
발생 트래픽 분포	$\lambda = B * \alpha$ 인 포아송 분포로 발생
총 발생 패킷수	5,000 packets
EI-CER 지연시간	Assured Voice = 35 ms Assured Multimedia Conferencing = 35 ms Short Message = 50 ms Non-Assured Voice = 35 ms Broadcast Video = 50 ms
target delay 최솟값	5 ms
전파 지연시간	100 ms

처리한다. 2순위(Lowest priority)는 3가지 응용 서비스(Short Message, Non-Assured Voice, Broadcast Video)를 나중 처리한다. 설계한 구조를 바탕으로 서

표 5. target delay와 interval rate 최적화 성능분석 결과
Table 5. Performance analysis results of target delay and interval rate

		target delay (ms)	interval rate 지수	폐기된 패킷 수 (pkt)	큐 안정화 시점 (pkt)	outdated 패킷 수 (pkt)
기존 방식	Assured Voice	5	0	1,492	5,000	1,594 (31.88 %)
	Assured Multimedia Conferencing	5	0	2,045	4,997	1,765 (35.30 %)
	Short Message	5	0	1,292	4,963	1,346 (26.92 %)
	Non-Assured Voice	5	0	1,668	4,909	1,866 (37.32 %)
	Broadcast Video	5	0	2,253	4,997	1,949 (38.98 %)
제안 방식	Assured Voice	5	0.4	1,651 (+159)	1,845 (-3,155)	599 (11.98 %)
	Assured Multimedia Conferencing	5.18	1.0	2,217 (+172)	869 (-4,128)	357 (7.14 %)
	Short Message	5	0.3	1,445 (+153)	2,574 (-2,389)	662 (13.24 %)
	Non-Assured Voice	5.82	0.7	1,891 (+223)	1,117 (-3,792)	452 (9.04 %)
	Broadcast Video	6.64	1.0	2,381 (+128)	1,125 (-3,872)	375 (7.50 %)

비스 플로우 별로 논리적인 큐를 생성하고, 각 큐마다 알고리즘 1을 통해 얻어낸 파라미터를 적용한 FQ-CoDel 프로토콜을 구현한다. 시뮬레이션은 MATLAB으로 수행하고, 최적화는 CVX 툴을 이용한다. 각 응용 서비스에 해당하는 플로우는 DWRR으로 동작하고 마지막에는 PQ(Priority Queuing)가 결합한 복합 스케줄러 기술을 적용하였다. 성능분석을 위한 추가 파라미터는 표 4와 같다. 각 응용 서비스에서는 병목구간 대비 1.3 ~ 1.8 배인 포아송 분포로 트래픽이 발생함을 가정한다. 병목구간에 진입되는 패킷의 평균 크기는 1,000 bytes로 가정한다^{8,9)}. 플로우마다 총 진입되는 패킷의 수는 5,000 개로 제한한다. 트래픽의 양은 패킷 단위로 확인하고, 패킷이 진입되고 처리되는 시간 ms 단위로 확인한다. EI-CER 지연시간은 각 서비스에서 하나의 플로우에 대한 값을 나타낸다¹⁰⁾.

4.2 성능분석 결과

표 5는 알고리즘 1에서 첫 번째 단계를 완료한 결과이다. 기존방식은 target delay와 interval rate 지수가 각각 5 ms과 0으로 고정된 값을 사용하였고, 제안 방식은 응용 서비스 및 네트워크 환경에 따라 optimal target delay와 optimal interval rate 지수를 적용하였다. 세 가지 성능지표의 정의는 다음과 같다.

- 폐기된 패킷 수: 개별 큐에 진입된 5,000 개의 패킷 중 마지막 패킷이 큐를 벗어나기까지 손실되는 패킷의 총 개수
- 큐 안정화 시점: 향후 모든 패킷이 목표 지연시간 이내 도달하는 시점을 안정화 시점으로 판단하며 목표 지연시간을 초과하게 되는 마지막 패킷의 인덱스

- outdated 패킷 수: 목표 지연시간 이후 수신 단계 도착하여 서비스 요구사항을 만족시키지 못한 패킷의 총 개수

기존방식과 비교하여 제안방식은 두 가지 특징이 있다. 첫째, 제안방식의 target delay는 다소 증가하여 허용 가능한 큐의 길이 β_{target} 를 증가시킨다. 둘째, 제안방식은 interval 간격을 적절하게 앞당기게 되어 더 잦은 시간마다 큐를 체크한다. 종합적으로 제안방식은 빈번한 시간마다 패킷을 폐기하지만, 적절한 β_{target} 를 기준으로 패킷을 폐기한다. 따라서 전체 폐기되는 패킷의 양은 다소 증가하나, 큐 안정화 시점을 매우 앞당겨 주고 outdated 패킷의 수를 감소시키는 장점이 있다. 다시 말해서 제안방식은 큐 혼잡으로 인해 β_{target} 를 초과하여 필수적으로 손실이 필요한 패킷만 폐기했음을 알 수 있다. 이러한 결과는 다섯 가지 응용 서비스 모두 공통으로 나타난다.

그림 2와 그림 3은 다섯 가지 응용 서비스 중 Non-Assured Voice 서비스의 기존방식과 제안방식을 비교한 결과이다. 그림 2는 시간에 따른 큐 내부 패킷의 양을 나타내고, 그림 2에서 파란 막대는 패킷이 폐기되는 양을 나타낸다. 빨간 실선은 패킷이 처리되거나 폐기되기 이전, 검정 실선은 이후 패킷의 양을 나타낸다. 기존방식은 그래프의 fluctuation이 심해 큐 길이가 안정되지 못하지만, 제안방식은 약 300 ms 이후 어느 정도 saturation 상태임을 알 수 있다. 총 패킷

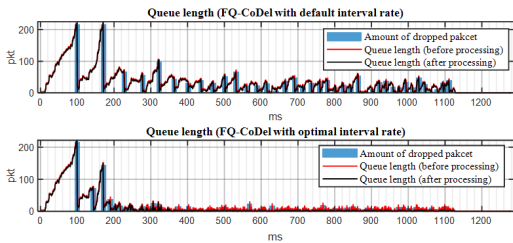


그림 2. 큐 내부 패킷의 양 (상:기존방식, 하:제안방식)
Fig. 2. Queue length (Top:Conventional, Bottom:Proposed)

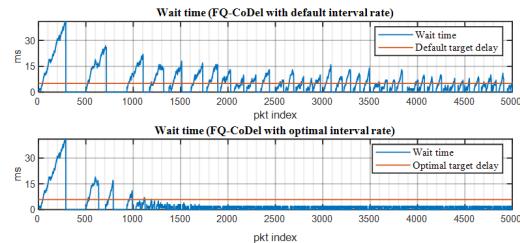


그림 3. 패킷 별 대기시간 (상:기존방식, 하:제안방식)
Fig. 3. Wait time (Top:Conventional, Bottom:Proposed)

표 6. 1순위 서비스 quantum 최적화 성능분석 결과
Table 6. Performance analysis results of quantum optimization for 1st level priority service

		기존방식		제안방식	
		default quantum	round	optimal quantum	round
Assured Voice	1	1,514	9,288	1,162	12,101
	2	1,514	9,288	1,162	12,101
	3	1,514	9,288	1,162	12,101
	4	1,514	9,288	1,162	12,101
Assured Multimedia Conferencing	1	1,514	9,372	4,488	3,162
	2	1,514	9,372	1,340	10,588
	⋮				
	10	1,514	9,372	1,340	10,588
quantum sum		21,196		21,196	

표 7. 2순위 서비스 quantum 최적화 성능분석 결과
Table 7. Performance analysis results of quantum optimization for 2nd level priority service

		기존방식		제안방식	
		default quantum	round	optimal quantum	round
Short Message	1	1,514	9,155	1,876	7,389
	⋮				
Non-Assured Voice	16	1,514	9,155	1,876	7,389
	⋮				
Broadcast Video	1	1,514	9,238	1,877	7,452
	⋮				
last round	22	1,514	9,238	1,877	7,452
	⋮				
last round	1	1,514	9,337	1,887	7,492
	⋮				
last round		9,337 (s.t.: 91.1)		7,389 (s.t.: 51.9)	

폐기량은 제안방식이 기존방식보다 약 200 개 정도 증가한다. 그림 3은 패킷 별 대기시간을 나타낸다. 대기시간은 패킷이 큐에 진입되는 시점과 처리되어 나가는 시점의 시간 차이를 의미한다. 기존방식은 패킷을 적절한 시점에 폐기하지 못해 약 37 %의 패킷이 목표한 대기시간에 도달하지 못한다. 반면, 제안방식은 약 1,000번째 패킷 이후 모든 패킷이 안정적인 대기시간에 도달한다.

표 6과 표 7은 알고리즘 1에서 두 번째 단계를 완료한 결과이다. 기존방식은 quantum이 1,514 bytes로 고정된 값을 사용하였고, 제안방식은 optimal quantum을 사용한다. 표 6은 1순위에서 Assured Multimedia Conferencing 응용 서비스의 첫 번째 플로우를 최우선 처리하기 위해 최적화 수식 (10)을 적용하였다. 모든 플로우에서 필요한 quantum의 합은 유지하면서 첫 번째 플로우의 지연시간은 1/3 수준으로 감소시킨다. 표 7은 2순위에서 전체 응용 서비스의 처리 지연시간을 감소시키기 위해 최적화 수식 (11)을 적용하였다. 50 개의 플로우에서 마지막 패킷이 처리

되기까지 필요한 지연시간을 감소시키고, 응용 서비스 간 지연시간의 차이를 최소화하여 표준편차가 크게 감소한다.

V. 결 론

본 연구에서는 큐 관리 및 스케줄러 기법이 결합된 FQ-CoDel 프로토콜을 다계층 통합 전송 백본망에 적용하기 위한 기술을 제안하였다. 다양한 응용 서비스에서 전송 효율성을 얻기 위하여 개별 응용 서비스의 QoS를 보장하고 네트워크 혼잡제어를 관리하기 위하여 FQ-CoDel 매개변수를 적응적으로 최적화하는 알고리즘을 설계하였다. 제안하는 큐 관리 방식은 큐 혼잡 상황 발생 시, 적절한 시점에서 패킷을 폐기함에 따라 최대한 많은 양의 패킷을 목표 지연시간 이내에 도달하게 하며 큐 안정화 시점을 앞당겨 준다. 특히, 음성, 비디오, 메시지 등 응용 서비스의 큐잉 지연 요구사항을 만족하기 위해 개별 큐를 동적으로 관리하는데 이점이 있다. 제안하는 스케줄링 방식은 작전 운용상 정보 중요도 및 군사 우선순위를 분류하여 우선 처리하기 위해 개별 플로우의 가중치를 변경하거나 지연시간 감소를 위해 전체 플로우의 가중치를 조정할 수 있는 장점이 있다.

References

[1] J. Kim, J. Yun, E. Jang, and S. Choi, "A study on adaptive queue management algorithm considering characteristics of multimedia," *J. Inst. Electron. and Inf. Eng.*, vol. 48, no. 5, pp. 140-148, Nov. 2011.

[2] K. Nichols, V. Jacobson, A. McGregor, Ed., and J. Iyengar, Ed., "Controlled Delay Active Queue Management," RFC 8289, Jan. 2018.

[3] K. Nichols and V. Jacobson, "Controlling queue delay," *ACM Queue*, vol. 10, no. 5, May 2012.

[4] Dipesh M. Raghuvanshi, Annappa B., and Mohit P. Tahiliani, "On the effectiveness of codel for active queue management," in *Proc. ICACCT*, Apr. 2013.

[5] T. Hoeiland-Joergensen, P. McKenney, D. Taht, J. Gettys, and E. Dumazet, "The flow queue CoDel packet scheduler and active queue management algorithm," RFC 8290,

Jan. 2018.

[6] E. Grigorescu, C. Kulatunga, and G. Fairhurst, "Evaluation of Priority Scheduling and Flow Starvation for Thin Streams with FQ-CoDel," in *Proc. EuCNC*, Jun. 2015.

[7] C. Kulatunga, N. Kuhn, G. Fairhurst, and D. Ros, "Tackling bufferbloat in capacity-limited networks," in *Proc. EuCNC*, Jun. 2015.

[8] J. Ye, K.-C. Leung, and Victor O.K. Li, "Optimal delay control for combating bufferbloat in the internet," in *Proc. IEEE ICCS*, Dec. 2016.

[9] Mobin, S. Momen, and N. Mohammed, "A packet level simulation study of adhoc network with network simulator-2," in *Proc. ICEEICT*, Mar. 2017.

[10] UCR_2013_Combined_signed, The Office of the DoD Chief Information Officer, Jan. 2013.

정 소 이 (So-Yi Jung)



2009년~2013년 : 아주대학교
전자공학과 공학사
2013년~2015년 : 아주대학교
전자공학과 공학석사
2015년~2016년 : 한국화학융합
시험연구원 주임연구원
2016년~현재 : 아주대학교 전자
공학과 박사과정

<관심분야> 군 전송통신, 셀룰러 차량통신 MAC,
NB-IoT 랜덤 액세스, 소형셀 캐싱
[ORCID:0000-0001-8435-0646]

이 충 희 (Choong-Hee Lee)



1999년~2006년 : 아주대학교
전자공학과 공학사
2006년~2008년 : 아주대학교
전자공학과 공학석사
2008년~2015년 : 아주대학교
전자공학과 공학박사
2015년~현재 : LIG넥스원 C4I
연구소 선임연구원

<관심분야> 전술통신, 이동통신, QoS, QoE
[ORCID:0000-0002-7808-8554]

권 대 훈 (Dae-Hoon Kwon)



1995년~1999년 : 경북대학교 공
학사
2000년~2002년 : 경북대학교 공
학석사
2002년~현재 : 국방과학연구소
선임연구원
<관심분야> 군 전술통신,
MANET, 무선통신

[ORCID:0000-0003-0304-5530]

박 형 원 (Hyung-Won Park)



1999년~2003년 : 아주대학교 정
보 및 컴퓨터공학과 공학사
2003년~2005년 : 아주대학교 정
보 통신공학과 공학석사
2005년~2009년 : 아주대학교 정
보 통신공학과 공학박사
2009년 9월~현재 : LIG넥스원
C4I연구소 수석연구원

<관심분야> 전술데이터링크, 군 위성통신, 이동통신
[ORCID:0000-0002-5139-6450]

김 재 현 (Jae-Hyun Kim)



1987년~1996년 : 한양대학교 전
산과 학사 및 석/박사 졸업
1997년~1998년 : 미국 UCLA
전기전자과 박사 후 연수
1998년~2003년 : Bell Labs, NJ,
USA, 연구원
2003년~현재 : 아주대학교 전자
공학부 교수

<관심분야> QoE/QoS, 무선 MAC 프로토콜, IEEE
802.11/15, B5G 통신 시스템, 국방 전술네트워크,
위성시스템 등
[ORCID:0000-0003-4716-6916]