

<https://doi.org/10.7236/JIIBC.2020.20.1.107>
JIIBC 2020-1-15

통합 플로우 기반 네트워크의 지연시간 최대치 분석

Delay Bound Analysis of Networks based on Flow Aggregation

정진우*

Jinoo Joung*

요약 본 연구에서는 IEEE 802.1 time sensitive network(TSN) task group(TG)에서 표준화 중인 asynchronous traffic shaping (ATS) 기술에서 제시된 minimal interleaved regulator(IR) 개념을 확장 적용한 통합 플로우 기반 지연시간 보장 프레임워크를 분석하였다. 해당 프레임워크는 단위 네트워크의 출력포트에 IR을 적용하여 burst 축적을 방지하면서, 동시에 단위 네트워크 안에서는 입출력 포트를 기준으로 플로우를 통합하여 복잡도를 낮출 수 있다. 본 연구에서는 다양한 파라미터들 가진 네트워크에서의 수치적 분석을 통해서, 제안된 낮은 복잡도의 프레임워크의 성능이 기존 integrated services (IntServ) 프레임워크보다 더 우수하거나 비슷함을 보였다. 특히 통합 플로우의 크기와 단위 네트워크의 크기가 클수록 성능이 우수해짐을 확인하였다.

Abstract We analyze the flow aggregate (FA) based network delay guarantee framework, with generalized minimal interleaved regulator (IR) initially suggested by IEEE 802.1 time sensitive network (TSN) task group (TG). The framework has multiple networks with minimal IRs attached at their output ports for suppressing the burst cascades, with FAs within a network for alleviating the scheduling complexity. We analyze the framework with various topology and parameter sets with the conclusion that the FA-based framework with low complexity can yield better performance than the integrated services (IntServ) system with high complexity, especially with large network size and large FA size.

Key Words : Delay guarantee, TSN, flow aggregate, interleaved regulator

1. 서론

네트워크 지연시간 (network delay) 보장을 위한 국제 표준으로 IEEE 802.1 TSN^[1]과 IETF deterministic network (DetNet)^[2]이 대표적이다. 이들 표준은 단일 도메인 네트워크(single domain network)에서의 지연시간 보장을 목표로 한다. TSN에서 제시된 asynchronous traffic shaping (ATS)^[3] 기술은 interleaved

regulator (IR)를 사용하여 입력포트별, 클래스별 트래픽 레귤레이션을 출력 포트의 시작점에서 수행한다. 그림 1은 이를 도시한 것이다. Interleaved regulator와 strict priority 기반 클래스별 FIFO 시스템이 나란히 구현되어 있다. 이러한 regulation 기능을 적절히 활용하면 FIFO 스케줄러를 통과하면서 플로우가 통합 분리를 반복함에 따라 발생하는 burst 축적 현상을 방지할 수 있다^[4].

*정회원, 상명대학교 휴먼지능정보공학과
접수일자: 2019년 12월 7일, 수정완료: 2020년 1월 7일
게재확정일자: 2020년 2월 7일

Received: 7 December, 2019 / Revised: 7 January, 2020 /

Accepted: 7 February, 2020

*Corresponding Author: jjoung@smu.ac.kr

Dept of Human-centered AI, Sangmyung Univ., Seoul, Korea

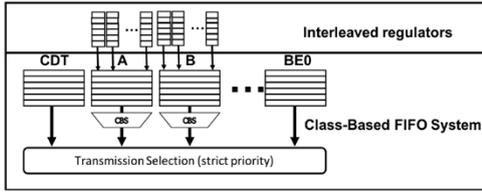


그림 1. TSN의 ATS: IR + Class 기반 FIFO 시스템^[3]
 Fig. 1. TSN ATS: IR + Class based FIFO system^[3]

최근 interleaved regulator를 대규모 네트워크에 적용하는 프레임워크가 제시되었다^[5]. 해당 연구에서는 interleaved regulator가 FIFO 시스템의 지연시간 최대치를 늘이지 않는다는 이론을 확장하여, FIFO 특성이 개별 플로우에만 적용되는 시스템에서도 해당 이론이 만족함을 증명하였다^[5]. 더 나아가 이러한 “플로우 FIFO 시스템”이 단위 네트워크일 수 있다는 것을 제안하고 IR을 단위 네트워크의 출력포트에 구현하며, 단위 네트워크에서 플로우를 통합하여 스케줄러의 복잡도를 크게 낮추는 방안을 제시하였다^[4]. 본 연구에서는 다양한 파라미터를 가진 네트워크에서의 수치적 분석을 통해서, 제안된 낮은 복잡도를 가진 프레임워크의 성능이 기존의 플로우 기반 integrated services (IntServ) 프레임워크보다 더 우수할 수 있음을 보였다. 특히 통합 플로우의 크기와 단위 네트워크의 크기가 클수록 성능이 우수해짐을 확인하였다.

II. 통합플로우 기반 프레임워크

Le Boudec^[6]은 아래 나열된 조건들이 만족되면 그림 1의 interleaved regulator가 추가된 TSN ATS시스템의 지연시간 최대치가 class-based FIFO system만의 지연시간 최대치와 동일하다는 것을 증명하였다.

- 1) FIFO 시스템으로 진입하는 모든 플로우는 잘 알려진 Leaky-bucket 등의 레귤레이터를 통과한 것과 같은 진입 특성을 가져야 한다. 즉 평균 진입 속도와 최대 burst 크기에 제한이 있어야 한다.
- 2) FIFO 시스템은 모든 packet들을 FIFO로 출력한다.
- 3) IR은 모든 플로우를 FIFO 시스템으로의 진입 특성을 재현하도록 regulate 한다. 즉, regulator의 특성 파라미터 (예를 들어 평균 진입 속도와 최대 burst 크기)를 진입 시와 동일하게 만든다.

4) (Minimal IR) IR은 Head of Queue의 packet이 출력조건을 만족하는 순간 즉시 전송한다. 이를 Minimal regulator라고 한다.

5) IR은 출력조건을 만족하는 packet들에 대해서 zero 지연시간을 제공할 수 있다. 예를 들어 queue가 비어 있는 상황에서 packet이 들어오면 이를 cut-through 할 수 있다.

한 가지 주목할 점은, 위와 같은 조건을 모두 만족하여서 지연시간 최대치가 늘어나지 않는다고 해도, 이는 전체 시스템의 최대 지연시간에 대해서만 성립하는 특성이라는 것이다. 즉, 특정 플로우의 지연시간 최대치는 증가할 수 있다. [5]에서 위의 조건 중 2번을 완화하여 좀 더 일반적인 상황에서도 위의 지연시간 최대치가 증가하지 않는다는 theorem이 만족함을 보였다.

Theorem^[5]: 개별 플로우의 패킷들에 대해서만 FIFO를 보장하는 시스템을 “플로우 FIFO 시스템” S라 하자. S와 Minimal IR을 합친 전체시스템의 최대 지연시간은 시스템 S만의 최대 지연시간과 동일하다.

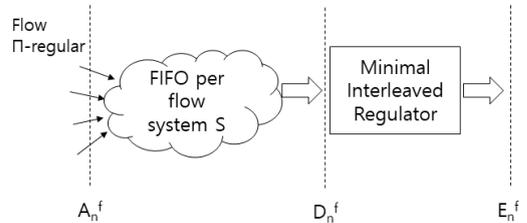


그림 2. 플로우 FIFO 시스템과 Minimal IR의 조합
 Fig. 2. Combination of the flow FIFO system and the minimal interleaved regulator

이러한 theorem의 확장으로 플로우 간의 경로 차이로 플로우 간 패킷 출력 순서가 바뀌는 경우에도 IR의 사용과 theorem 적용이 가능하게 된다. 따라서 다양한 우선순위를 가진 플로우들이 진입되는 다양한 경로를 가진 하나의 네트워크에도 IR을 적용할 수 있다. 또한, S에서 증가한 burst size가 IR을 통과하면서 완화되어 원래 burst size 값으로 회귀하는 특성을 이용해 다음과 같은 단대단 지연시간 보장 프레임워크를 설계할 수 있다.

- 플로우는 높은 우선순위와 낮은 우선순위 두 종류로 나눈다.
- 낮은 우선순위 플로우들은 모든 노드의 출력포트에서 하나의 FIFO queue에 집어넣고 preemption이 있는 strict priority 방식으로 처리한다.
- 높은 우선순위 플로우들은 아래와 같이 처리한다.

- 적당한 크기의 단위 네트워크를 선택하여 네트워크의 경계(boundary)마다 IR을 설치한다.
- 단위 네트워크의 입력으로 Arrival curve를 따르는 최초 인입 플로우와 IR에서 출력되는 플로우만을 허용한다.
- 단위 네트워크를 통과하는 플로우 중 네트워크 입력포트와 네트워크 출력포트가 동일한 플로우들을 하나의 통합 플로우(Flow aggregate, FA)로 묶는다.
- 네트워크의 중계노드에서는 FA별로 queue를 할당하여 fair queuing 방식의 스케줄링을 수행한다. 이렇게 함으로써 모든 높은 우선순위 FA의 최대 지연 시간을 보장할 수 있다. 네트워크 경계에서의 새로운 플로우들 간 통합으로 인해 burst 크기가 증가하지만 IR을 통과하면서 원래 값을 되찾는다.

III. 제안된 프레임워크의 성능 분석

하나의 플로우 i 가 네트워크를 지나면서 latency-rate (LR) 서버들만을 통과한다면, 이 플로우 i 의 패킷들이 겪는 단대단 지연시간의 최대치는 다음과 같은 식으로 표현된다^[7].

$$D_i \leq \frac{\sigma_i - L_i}{\rho_i} + \sum_{j=1}^k \Theta_i^{S_j} \quad (1)$$

아래에서 자주 사용하는 수학 기호에 대해서 표로 정리하였다.

표 1. 수학 기호와 그 의미

Table 1. Notations and their meaning

Notation	Meaning
L_i	Max packet length of flow i
r	Link capacity
σ_i	Max burst size of flow i
ρ_i	Input data rate of flow i
ϕ_i	Quantum value assigned for flow i
$\Theta_i^{S_j}$	Latency of flow i at sever S_j
D_i	Delay experienced by packets of flow i

가장 이상적인 scheduler라고 할 수 있는 packetized generalized processor sharing(PGPS)과 간단한 라운드 로빈 기반의 deficit round robin(DRR)^[8]이 LR 서버에 포함된다. PGPS의 latency는 다음과 같다.

$$\Theta_i^{PGPS} = \frac{L_i}{\rho_i} + \frac{L_{\max}}{r}. \quad (2)$$

퀀텀(quantum) 값이 패킷의 최대길이보다 작은 경우를 포함하는, 일반적인 DRR의 latency는 다음과 같이 주어진다^[9].

$$\Theta_i^{DRR} = \frac{1}{r} \left[(F - \phi_i) \left(1 + \frac{L_i}{\phi_i} \right) + \sum_{n=1}^N L_n \right]. \quad (3)$$

여기서 F 는 모든 플로우의 퀀텀 값 ϕ_i 들의 합이다. 퀀텀은 플로우 별로 할당된 서비스 rate과 비례하여 결정되는, 한 번에 서비스되는 데이터양을 의미한다^[8].

1. Symmetric 네트워크 case

그림 3과 같이 모든 플로우가 동일한 특성을 가지며, 동일하게 h 홉을 거치는 네트워크를 고려하자. 모든 노드는 2개의 입력과 2개의 출력포트가 있다. n^{2^h} 개의 플로우가 입력포트로 인입되며 이 중 $n^{2^{h-1}}$ 개가 같은 포트 로 출력된다. 두 번째 노드에서는 이 중 $n^{2^{h-2}}$ 개가 같은 포트 로 출력되며 마지막 노드에서는 이 중 $n^{2^{h-h}}=n$ 개가 같은 포트 로 출력된다. 따라서, 같은 {입력, 출력포트} 쌍을 가지는 플로우는 n 개이다. 이러한 입출력 패턴이 모든 노드에서 일어난다고 하자.

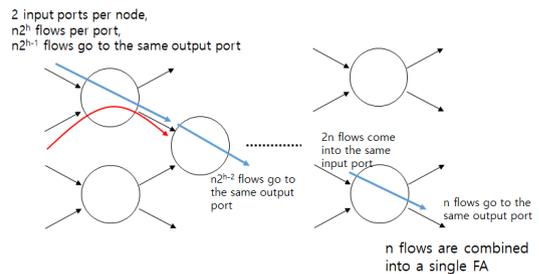


그림 3. 분석에 사용한 generic 단위 네트워크 구조

Fig. 3. Architecture of the single generic network used for the analysis

가. PGPS 스케줄러 case

먼저, 개별 플로우 기반 스케줄링 방식에서의 지연시간을 구하면, PGPS 스케줄러인 경우는

$$\Theta_i^{PGPS} = \frac{L_i}{\rho_i} + \frac{L_{\max}}{r} = \frac{L}{r/n^{2^h}} + \frac{L}{r} = \frac{(n^{2^h} + 1)L}{r}.$$

편의상 $\sigma_i = L$ 이라 하면 (1)에 의해서

$$D_i \leq \sum_{j=1}^h \Theta_i^{S_j} = \frac{h(n^{2^h} + 1)L}{r}.$$

통합플로우 기반 PGPS 스케줄링 방식에서는 2^h 개의 통합플로우가 존재하므로 (2)에 의해

$$\Theta_i^{PGPS} = \frac{L_i}{\rho_i} + \frac{L_{\max}}{r} = \frac{L}{r/2^h} + \frac{L}{r} = \frac{(2^h + 1)L}{r} \text{ 이며}$$

burst가 nL 이므로,

$$D_i \leq \frac{\sigma_i - L_i}{\rho_i} + \sum_{j=1}^h \Theta_i^S = \frac{\{(h+n-1)2^h + h\}L}{r}.$$

개별 플로우 기반과 통합 플로우 기반 지연시간 최대치의 차이는 $\frac{(h-1)(n-1)2^h L}{r}$. 즉 통합 플로우의 크기, 최대 패킷 크기에는 선형적으로 비례하고, 단위 네트워크의 크기에는 기하급수적으로 비례하며, 링크의 용량에는 반비례한다. $n=1$ 이거나 $h=1$ 일 때 차이가 없다.

나. DRR 스케줄러 case

개별플로우 기반 스케줄링 방식에서 DRR을 사용하는 경우를 고려해보자. 편의상 $\phi_i = L_i = L$ 이라고 하자. 단일 노드에서의 latency가 (3)에 의해

$$\Theta_i^{DRR} = \frac{2(n2^h - 1)L + n2^h L}{r} = \frac{(3n2^h - 2)L}{r} \text{ 이므로 전체}$$

$$\text{네트워크의 지연시간은 } D_i \leq \sum_{j=1}^h \Theta_i^S = \frac{h(3n2^h - 2)L}{r}.$$

통합 플로우 기반 DRR 스케줄링 방식에서도 모든 FA가 동일한 서비스 rate이므로 $\phi_i = L_i = L$ 로 설정할 수 있다. 다만 여기서 FA의 수가 2^h 인 것에 유의하자.

$$\Theta_i^{DRR} = \frac{(3 \cdot 2^h - 2)L}{r} \text{ 이며, burst 크기는 } nL \text{ 이므로 } D_i \leq \frac{((3h+n-1)2^h - 2h)L}{r}.$$

$$\text{두 지연시간 최대치의 차이는 } \frac{(3h-1)(n-1)2^h L}{r}.$$

즉 통합 플로우의 크기, 최대 패킷 크기에는 선형적으로 비례하고, 단위 네트워크의 크기에는 기하급수적으로 비례하며, 링크의 용량에는 반비례한다. PGPS의 경우보다 지연시간 최대치의 차이가 3배가량 더 벌어진다. Symmetric 환경에서는 더 나아가 이러한 단위 네트워크가 연속되어도 성능 차이가 계속 선형적으로 벌어질 수 밖에 없다.

2. Asymmetric 네트워크 case

이번에는 플로우들의 특성이 동일하지 않은 경우에 지연시간이 어떻게 변화하는지 알아보자. 우리는 다른 플로우의 지연시간 최대치가 상대적으로 커서 IR을 통과하면서 관찰대상 플로우의 지연시간 최대치가 더 커지는 경

우를 분석하고자 한다. 플로우의 상대적인 전송속도(rate)가 작을수록 단위 네트워크에서의 지연시간이 커진다는 점을 활용한다. 따라서 그림의 관찰대상 플로우와, 이 플로우와 통합되는 $(n-1)$ 개의 플로우만 2배의 상대적인 전송속도를 요구해서 할당받는다 고 하자. 이러한 단위 네트워크가 연결되어 그림 4와 같이 전체 네트워크를 구성한다고 하자. 단위 네트워크의 모든 환경은 동일하다. 관찰 대상인 플로우는 d 개의 단위 네트워크를 경유한다.

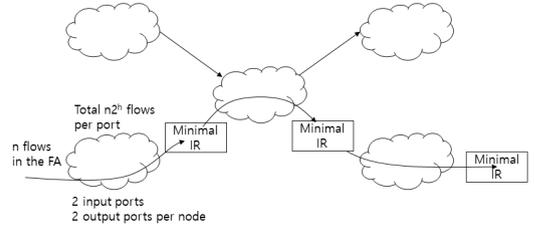


그림 4. 복수의 단위 네트워크를 포함한 전체 네트워크 구조
Fig. 4. Architecture of the multiple networks used for the analysis

가. PGPS 스케줄러 case

개별 플로우 기반 스케줄링 방식에서의 관찰대상 플로우 i 의 지연시간을 구하면, PGPS 스케줄러인 경우는

$$\Theta_i^{PGPS} = \frac{(n2^h + n + 2)L}{2r} \text{ 이며 플로우의 burst가 } L \text{이}$$

기 때문에 $D_i^{FA} \leq \sum_{j=1}^h \Theta_i^S = \frac{h(n2^h + n + 2)L}{2r}$. 전체 네트워크를 지나면서 경유하는 단위 네트워크의 수를 d 라고 하면, 플로우 기반 PGPS 시스템에서 전체 네트워크의 지연시간 T_i^F 은

$$T_i^F \leq \max(dD_i^{FA}) = \frac{dh(n2^h + n + 2)L}{2r}.$$

통합플로우 기반 PGPS 스케줄링 방식에서는 관찰대상 FA가 받는 서비스 rate이 $2nr/(n2^h+n)$ 이므로

$$\Theta_i^{PGPS} = \frac{(2^h + 3)L}{2r} \text{ 이며 통합플로우의 burst가 } nL \text{이}$$

기 때문에

$$D_i^{FA} \leq \frac{(n+h-1)2^h + (3h+n-1)L}{2r/L}.$$

따라서 지연시간 최대치의 차이는

$$\frac{(h-1)(n-1)(2^h + 1)L}{2r}. \text{ 여기서도 } h=1 \text{이거나 } n=1 \text{인}$$

경우 차이가 없음을 유의하자.

한편 같은 네트워크 시나리오에서 관찰대상 플로우의 1/2의 서비스 rate을 할당받은 플로우의 지연시간 최대치를 구해보자. 이 플로우를 j 라고 하자. FA기반 시스템

$$\text{에서 } j \text{의 PGPS latency는 } \theta_j^{PGPS} = \frac{(2^h + 2)L}{r}$$

$$\text{이며 } D_j^{FA} \leq \frac{(n+h-1)2^h + (2h+n-1)}{r/L}.$$

여기서 i 와 j 의 최대 지연시간 차이는

$$\max D_j^{FA} - \max D_i^{FA} = \frac{(n+h-1)(2^h+1)}{2r/L}.$$

항상 $\max D_j^{FA} \geq \max D_i^{FA}$ 이며 D_i^{FA} 가 단위 네트워크에서 임의의 플로우가 겪는 지연시간의 최대치이다. 따라서 마지막 네트워크의 출력포트 이후에도 Minimal IR을 경유한다면 i 가 겪는 전체 네트워크의 지연시간 T_i^{FA} 는

$$T_i^{FA} \leq \max(dD_j^{FA}) = \frac{(n+h-1)(2^h+1)+h}{r/dL}.$$

두 최대치의 차이를 FA-gain이라고 정의하자. 아래와 같이 주어진다.

$$\begin{aligned} FAgain^{PGPS} &= \max(T_i^F) - \max(T_i^{FA}) \\ &= \frac{(h-2)(n-2)(2^h+1) - 2(2^h+h)}{2r/dL}. \end{aligned}$$

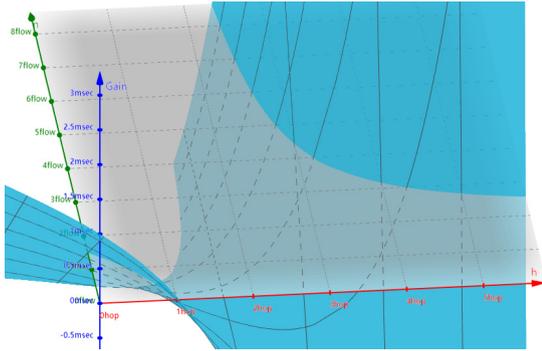


그림 5. $r=1G$, $L=10000\text{bit}$, $d=3\text{일}$ 때 PGPS FA gain
 Fig. 5. PGPS FA gain with $r=1G$, $L=10000\text{bit}$, $d=3$

그림 5는 $FA\text{-gain}^{PGPS}$ 를 도시한 것이다. h 와 n 이 모두 4 이상일 때부터 양의 값을 가지며 두 값이 커질수록 큰 gain을 보인다.

나. DRR 스케줄러 case

다음으로 DRR을 사용하는 전체 네트워크에서의 플로우 시스템과 FA 시스템을 비교하자. 편이상 $\phi_i = 2L_i = 2L$

이라고 하자. i 와 FA시스템에서 통합되는 $(n-1)$ 개의 플로우의 쿼터 값도 $2L$ 이며 이 n 개의 플로우들을 제외하면 모두 L 을 쿼터 값으로 가진다.

먼저 플로우 시스템의 latency와 delay를 계산하자. 단일 노드에서 i 의 latency는

$$\theta_i^{DRR} = \frac{5n2^h + 3(n-2)}{2r/L} \text{이며 플로우의 burst가 } L$$

이기 때문에 단위 네트워크의 지연시간은

$$D_i \leq \sum_{j=1}^h \theta_i^S = \frac{5hn2^h + 3h(n-2)}{2r/L}.$$

플로우 기반 시스템에서 전체 네트워크에서의 지연시간은

$$T_i^F \leq \max(dD_i^F) = \frac{5hn2^h + 3h(n-2)}{2r/dL}.$$

이번에는 FA 시스템에서 관찰대상 플로우 i 의 1/2의 rate을 할당받은 임의의 플로우 j 를 고려하자. j 의 단위 네트워크에서의 지연시간 최대치가 해당 단위 네트워크("플로우 FIFO 시스템")의 지연시간 최대치이다.

$$\theta_j^{DRR} = \frac{(2n+1)2^h + 2(n-1)}{r/L} \text{이며 } j \text{가 속한 통합}$$

플로우의 burst가 nL 이기 때문에 단위 네트워크에서 j 의 지연시간은

$$D_j^{FA} \leq \frac{(2hn+h+n-1)2^h + (2h+1)(n-1)}{r/L}.$$

$\max(D_j^{FA})$ 가 단위 네트워크에서 임의의 플로우가 겪는 지연시간의 최대치이며 따라서 i 가 전체 네트워크에서 겪는 지연시간 T_i^{FA} 는

$$\begin{aligned} T_i^{FA} &\leq \max(dD_j^{FA}) \\ &= \frac{(2hn+h+n-1)2^h + (2h+1)(n-1)}{r/dL}. \end{aligned}$$

FA로 인한 지연시간 최대치의 gain은

$$\begin{aligned} FAgain^{DRR} &= \max(T_i^F) - \max(T_i^{FA}) \\ &= \frac{(hn-2h-2n+2)2^h - hn+2h+2n-2}{2r/dL}. \end{aligned}$$

DRR의 경우도 충분히 큰 값의 h 와 n 에 대해서 gain이 positive 값을 가짐에 유의하자. 그림 6은 $FA\text{-gain}^{DRR}$ 을 도시한 것이다. h 가 3 이상, n 이 4 이상일 때부터 양의 값을 가지며 두 값이 커질수록 큰 gain을 보인다.

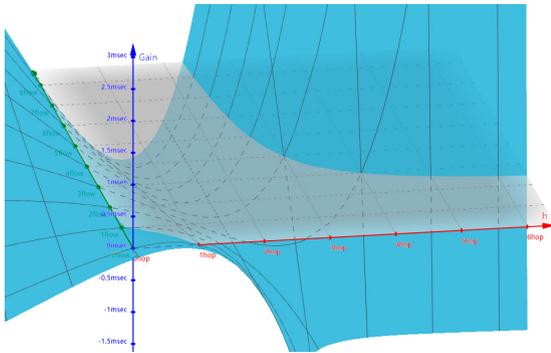


그림 6. $r=1G$, $L=10000\text{bit}$, $d=3\text{일}$ 때 DRR FA gain
 Fig. 6. DRR FA gain with $r=1G$, $L=10000\text{bit}$, $d=3$

IV. 결 론

본 연구에서는 Interleaved regulator를 단위 네트워크 사이에 배치하고 단위 네트워크에서는 통합 플로우 기반 스케줄링을 함으로써 대규모 네트워크에서 낮은 복잡도로 지연시간의 최대치를 보장하는 프레임워크를 분석하였다. 다양한 환경에서의 수치적 분석을 통해서, 제안된 낮은 복잡도의 프레임워크의 성능이 기존 integrated services (IntServ) 프레임워크보다 더 우수할 수 있음을 보였다. 특히 통합 플로우의 크기와 단위 네트워크의 크기가 클수록 성능이 우수해짐을 확인하였다. 통합 플로우와 단위 네트워크의 크기가 모두 4 이상이면 안정적으로 성능 향상을 기대할 수 있었다.

References

[1] IEEE 802.1 Time-Sensitive Networking Task Group Home Page, <http://www.ieee802.org/1/pages/tsn.html>

[2] IETF DetNet Working Group, <https://datatracker.ietf.org/wg/detnet/about/>

[3] J. Specht and S. Samii, "Urgency-based scheduler for time-sensitive switched Ethernet networks," in Proc. 28th Euromicro Conf. Real-Time Syst. (ECRTS), pp. 75-85, Jul. 2016. DOI: 10.1109/ECRTS.2016.27

[4] Jinoo Joung, "End-to-end delay guarantee in IEEE 802.1 TSN with Non-work conserving scheduler", The

Journal of The Institute of Internet, Broadcasting and Communication (IIBC), Vol 18, No. 6, Dec. 2018. DOI: 10.7236/IIBC.2018.18.6.121.

[5] Jinoo Joung, "Framework for Network Delay Guarantee based on Flow Aggregate and Interleaved Regulators.", The Journal of Korean Institute of Communications and Information Sciences, Mar. 2020.

[6] J-Y. Le Boudec, "A theory of traffic regulators for deterministic networks with application to interleaved regulators", IEEE/ACM Transactions on Networking Volume: 26, Issue: 6, Dec. 2018. DOI: 10.1109/TNET.2018.2875191

[7] Dimitrios Stiliadis and Anujan Varma. "Latency-Rate Servers: A General Model for Analysis of Traffic Scheduling Algorithms." IEEE/ACM Transaction on Networking, vol. 6, no. 5, Oct, 1998. DOI: 10.1109/90.731196

[8] M. Shreedhar and G. Varghese, "Efficient fair queueing using deficit round-robin," IEEE/ACM Transaction on Networking, vol. 4, no. 3, pp. 375-385, June 1996. DOI: 10.1109/90.502236

[9] L. Lenzini, E. Mingozzi, and G. Stea, "Tradeoffs between low complexity, low latency, and fairness with deficit round-robin schedulers," IEEE/ACM Transactions on Networking, vol. 12, Issue 4, August 2004. DOI: 10.1109/TNET.2004.833131

저 자 소 개

정 진 우(정회원)



- 1992년 2월 : KAIST 전기공학과
- 1994년 8월 : NYU EE Master
- 1997년 8월 : NYU EE Ph.D.
- 1997년 10월 ~ 2005년 2월 : 삼성전자 종합기술원
- 2005년 2월 ~ 현재 : 상명대학교 휴먼지능정보공학과 교수

※ 본 연구는 상명대학교 교내연구비를 지원받아 수행하였음. This research was supported by a Research Grant from Sangmyung University.