

통합 플로우 기반 스케줄링과 Interleaved Regulator를 이용한 네트워크 지연시간 보장 프레임워크

정진우*

Framework for Network Delay Guarantee with Interleaved Regulators and Flow Aggregate-Based Scheduling

Jinoo Joung*

요약

IEEE 802.1 time sensitive network (TSN) task group(TG)에서 표준화 중인 asynchronous traffic shaping (ATS) 기술의 핵심은 Class 기반의 FIFO 시스템에 minimal interleaved regulator(IR)를 연결하면 개별 플로우의 버스트 축적(burst cascade)을 방지하면서도 지연시간 최대치를 증가시키지 않는다는 이론이다. 본 연구에서는 이것을 확장하여 FIFO 특성이 개별 플로우에만 적용되는 시스템에서도 해당 이론이 만족함을 증명하였다. 더 나아가 Class 기반 FIFO 시스템이 노드 내부로 제한되는 데 비해 “플로우 FIFO 시스템”이 하나의 네트워크일 수 있다는 것을 제안하고 Minimal IR을 네트워크의 최종 출력포트에 구현하며, 네트워크 안에서 플로우를 통합하여 스케줄러의 복잡도를 크게 낮추는 방안을 제시하였다. 수치적 분석을 통해서 본 연구에서 제안한 낮은 복잡도의 통합 플로우 기반 시스템의 성능이 기존 integrated services (IntServ) 프레임워크에서 제시한 높은 복잡도의 개별 플로우 기반 시스템보다 우수한 것을 확인하였다.

Key Words : End-to-end delay guarantee, TSN, scheduler, flow aggregate, interleaved regulator

ABSTRACT

The key to the asynchronous traffic shaping (ATS) technology being standardized in IEEE 802.1 time sensitive network (TSN) task group (TG) is the theorem that a class-based FIFO system, attached with a minimal interleaved regulator (IR) does not increase maximum delay while suppress the burst cascade. In this work it is proved that the theorem also holds with a system that is FIFO only to individual flows. Based on the observation that this “flow FIFO system” can be a single network, while the class-based FIFO system is within a single node, it is further suggested that the networks with minimal IRs attached at their output ports, with flow aggregates (FAs) within a network, can guarantee end-to-end delay while drastically reduce the scheduler complexity. Numerical analysis shows that the suggested FA-based system with low complexity yields similar or better performance than the integrated services (IntServ) system with high complexity.

* 이 성과는 과학기술정보통신부가 지원한 ‘정보통신방송연구개발사업’으로 지원을 받아 수행되었음.(과제고유번호: 2018-0-00846).

• 상명대학교 휴먼지능정보공학과, Sangmyung University, Department of Human-centered AI, jjoung@smu.ac.kr, 정회원
논문번호 : 201911-309-B-RN, Received November 21, 2019; Revised December 21, 2019; Accepted December 23, 2019

I. 서 론

스마트 팩토리, 차량 간 통신, 차량 내 통신, 전원 음향 네트워크, 대규모 전력 제어망 등 다양한 응용 분야에서 단대단 네트워크 지연시간 (end-to-end network delay)에 대하여 수 msec에서 수초까지의 엄격한 제한을 요구하고 있다. 관련 국제 표준들도 속속 등장하고 있는데, IEEE 802.1 TSN^[1]과 IETF deterministic network (DetNet)^[2]이 대표적이다. 이들은 단일 도메인 네트워크 안에서 지연시간(delay) 보장 및 무손실의 확정적 서비스를 제공하는 기술표준을 추구한다.

TSN TG에서 제시된 asynchronous traffic shaping (ATS)^[3] 기술은 interleaved regulator (IR)를 채택하였다. IR은 입력포트별, 클래스별 트래픽 레귤레이션을 출력 포트의 시작점에서 수행한다. 그림 1은 이를 도시한 것이다. Interleaved regulator와 클래스별 FIFO 시스템이 구현되었다.

본 연구에서는 먼저, 기존 연구들에서 제시된 interleaved regulator가 FIFO 시스템의 지연시간 최대치를 늘이지 않는다는 이론을 확장하여, FIFO 특성이 개별 플로우에만 적용되는 시스템에서도 해당 이론이 만족함을 증명하였다. 더 나아가 이러한 “플로우 FIFO 시스템”이 하나의 네트워크일 수 있다는 것을 제안하고 IR을 네트워크의 출력포트에 구현하며, 네트워크 안에서 플로우를 통합하여 스케줄러의 복잡도를 크게 낮추는 방안을 제시하였다. 수치적 분석을 통해서 본 연구에서 제안한 낮은 복잡도의 통합 플로우 기반 시스템의 성능이 기존 integrated services (IntServ) 프레임워크에서 제시한 높은 복잡도의 개별 플로우 기반 시스템보다 우수한 것을 확인하였다.

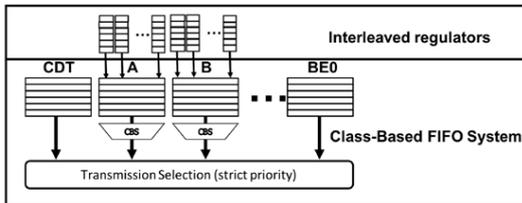


그림 1. TSN의 ATS: IR + Class 기반 FIFO 시스템[3]
Fig. 1. TSN ATS: IR + Class based FIFO system[3]

II. 관련 연구

Min-plus algebra를 바탕으로, 그림 1의 interleaved regulator가 추가된 TSN ATS시스템의 지연시간 최대

치가 기존의 class-based FIFO system의 지연시간 최대치와 동일하다는 것이 증명되었다^[4].

하나의 플로우의 패킷 sequence는 (A,L) 로 기술하는데, 여기서 $A=(A_1, A_2, \dots)$ 는 패킷 관찰시간의 sequence이며 $L=(L_1, L_2, \dots)$ 은 패킷 길이의 sequence이다. 예를 들어 해당 플로우의 두 번째 패킷의 관찰된 시간은 A_2 이며 길이는 L_2 이다. 시스템 S 에서 (A,L) 을 입력 sequence라고 하고 (D,L) 을 출력 sequence라고 했을 때 $A \leq D$ 라면 S 는 FIFO이다.

여기서 다음의 4가지 조건을 만족하는 function (혹은 operator) Π 에 대해서 고려하자.

- C1: Π 는 패킷 sequence (A,L) 을 받아서 시간 sequence로 변환한다.
- C2: Π 는 causal 하다. 즉 $\Pi(A,L) = A'$ 라면 A'_n 은 A_1, \dots, A_{n-1} 과 L_1, \dots, L_n 에 depend 하지만 A_m ($m \geq n$) 이나 L_m ($m \geq n + 1$)에 대해서는 independent 하다. A'_n 이 A_n 에 depend 하지 않다는 것에 주목하라.
- C3: Π 는 A 에 대해서 homogeneous 하다. 즉 임의의 상수 h 와 임의의 sequence A, L 에 대해서 $\Pi(A+h,L)=\Pi(A,L)+h$ 이다.
- C4: Π 는 A 에 대해서 isotone 하다. 즉, $A \leq A'$ 이면 임의의 sequence L 에 대해서 $\Pi(A,L) \leq \Pi(A',L)$ 이다.

Definition 1 (Pi-Regularity): C1-C4를 만족하는 함수 Π 에 대해서, $A \geq \Pi(A,L)$ 라면 플로우 패킷 sequence (A,L) 은 Π -regular이다.

여기서 $\Pi(A,L)$ 는 leaky bucket 등으로 강제하는 “arrival curve”로 해석할 수 있다. $A \geq \Pi(A,L)$ 라는 것은 패킷들이 모두 arrival curve보다 늦게 도착한다고 해석할 수 있다.

Theorem 1 (Minimal Π -Regulator): Π 가 C1-C4을 만족하는 함수라고 하자(Pi-regular operator). “Minimal Π regulator”는 인입 플로우 패킷 sequence (A,L) 을 출력 sequence (D,L) 로 변환하는 FIFO 시스템이며, 이때 $D_1=A_1$ 이고 $D_n = \max\{A_n, D_{n-1}, \Pi(D,L)_n\}$ 이다. 다음이 성립한다.

- 1) 이렇게 정의된 시스템은 해당 플로우에 대해서 Π -regulator이다.
- 2) (Minimality:) (A,L) 을 (D',L) 로 변환하는 임의의 Π -regulator와 모든 $n \in \mathbb{N}^+$ 에 대해서 $D'_n \geq D_n$.
- 3) 플로우 (A,L) 이 Π -regular라는 것과 $D=A$ 라는 것은 동치이다.

Minimal g-regulator^[5]와 Packetized Greedy

Shaper^[6] 등이 Minimal Π -Regulator의 좋은 예이다. 잘 알려진 Leaky bucket shaper는 Packetized greedy shaper의 특수한 경우이다.

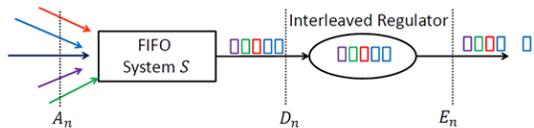
이제 여러 플로우가 혼합된 패킷 sequence (A, L, F) 를 고려한다. L_n 은 n 번째 패킷의 길이이다. F_n 은 n 번째 패킷이 속한 플로우이다. 모든 플로우가 각각 Π -Regular 하다고 하자. “Interleaved regulator (IR)”는 모든 플로우의 출력 sequence를 Π -Regular하도록 만들어 주는 FIFO 시스템이다. 이 과정에서 내부 패킷들의 전송이 지연될 수 있다. FIFO의 Head of Queue에 있는 패킷이 해당 플로우의 Π -Regular 조건에 맞을 때까지, queue 되어 있는 모든 패킷은 시스템 내에서 기다리게 된다. 한편 Minimal IR도 다음과 같이 정의할 수 있다.

Theorem 2: 패킷 sequence (A, L, F) 를 고려하자. 여기에 속한 플로우 별로 각각 Π^f -Regulator가 정의되어 있다고 하자. Minimal IR는 인입 패킷 sequence (A, L, F) 를 출력 sequence (D, L, F) 로 변환하는 FIFO 시스템이며, 이때 $D_1 = A_1$ 이고 $D_n = \max\{A_n, D_{n-1}, \Pi^{F_n}(D^{F_n}, L^{F_n})_{I(n)}\}$ 이다. 여기서 $I(n)$ 은 n 번째 패킷이 속한 플로우에서의 n 번째 패킷의 순번이다. 다음이 성립한다.

- 1) 이렇게 정의된 시스템은 패킷 sequence에 대해서 IR이다.
- 2) (Minimality:) (A, L, F) 을 (D', L, F) 로 변환하는 임의의 IR과 모든 $n \in \mathbb{N}^+$ 에 대해서 $D'_n \geq D_n$.
- 3) (A, L, F) 에 속한 모든 플로우가 Π -regular라는 것과 $D=A$ 라는 것은 동치이다.

Specht가 제시한 Urgency Based Scheduler^[7]가 Minimal IR의 대표적인 예이다.

Theorem 3: 패킷 sequence (A, L, F) 가 FIFO 시스템 S로 인입한다. 해당 sequence에 속하는 플로우 f 가 Π^f -Regular 하다. 출력 sequence (D, L, F) 는 minimal IR로 인입된다. 해당 Minimal IR은 플로우 f 에 대해서 Π^f 함수를 적용한다. 이러한 Minimal IR의 출력 sequence를 (E, L, F) 라고 하자. S와 Minimal IR을 합



Theorem: $\sup_n(D_n - A_n) = \sup_n(E_n - A_n)$

그림 2. [4]의 핵심 이론의 도식화
Fig. 2. Depiction of the key contribution of [4]

친 전체시스템의 최대 지연시간은 시스템 S만의 최대 지연시간과 동일하다. 즉 $\sup_{n \in \mathbb{N}^+}(D_n - A_n) = \sup_{n \in \mathbb{N}^+}(E_n - A_n)$. 그림 2는 이를 도식화한 것이다.

III. 제안하는 지연시간 보장 프레임워크

그림 2의 theorem이 성립하기 위해서 아래와 같은 몇 가지 조건이 만족 되어야 한다.

- 1) FIFO 시스템 S로 인입하는 모든 플로우는 Π -regular 해야 한다. 쉽게 설명하자면, 평균 인입 속도와 최대 버스트 크기에 제한이 있어야 한다. 즉 leaky-bucket 등의 레귤레이터를 통과한 것과 같은 인입 특성을 가져야 한다. 여기서 버스트는 짧은 시간 안에 인입이 허용되는 데이터의 최대량이다.
- 2) FIFO 시스템 S는 모든 인입 packet들을 FIFO로 출력한다.
- 3) IR은 모든 플로우를 기존 인입 특성을 재현하도록 regulate 한다. 즉, Π -regular의 특성 파라미터 (예를 들어 평균 인입 속도와 최대 버스트 크기)를 인입 시와 동일하게 만든다.
- 4) (Minimal IR) IR은 Head of Queue의 packet이 출력조건을 만족하는 순간 즉시 전송한다. 이는 Minimal regulator의 특성이다.
- 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 만족 하여서 지연시간 최대치가 늘어나지 않는다고 해도, 이는 전체 시스템의 최대 지연시간에 대해서만 성립하는 특성이라는 것이다. 즉, 특정 플로우의 지연시간 최대치는 증가할 수 있다. 만약 FIFO 시스템 전체의 최대지연시간이 D , 특정 플로우 f 의 최대지연시간이 D^f ($D > D^f$) 라면 IR을 적용한 후 해당 플로우의 최대 지연시간은 D 로 늘어난다.

본 연구를 통해 위의 조건 중 2번을 완화하여 좀 더 일반적인 상황에서도 위의 지연시간 최대치가 증가하지 않는다는 theorem이 만족함을 보였다. Theorem 2에서 정의된 IR은 패킷 sequence (A, L, F) 에 대해서 특별한 조건 없이 minimal 하다. (A, L, F) 의 개별 플로우 별로 Π^f -Regulator가 정의되어 있기에 하다면 충분하다. 따라서 위 조건 중 2번을 다음과 같이 변경시켜도 아래 Theorem 4가 성립한다.

수정된 2번 조건: 시스템은 개별 플로우의 패킷들

에 대해서만 FIFO를 보장한다. 이를 “플로우 FIFO 시스템”이라 하자.

Theorem 4: 하나 이상의 플로우 패킷 sequence (A^f, L^f) 가 “플로우 FIFO 시스템” S로 인입한다. 플로우 패킷 sequence f 는 Π^f -Regular 하다. 이들 플로우들의 출력 sequence (D, L, F) 는 minimal IR로 인입된다. 해당 Minimal IR은 플로우 f 에 대해서 Π^f 함수를 적용하여 regulate한다. 이러한 Minimal IR의 출력 sequence를 (E, L, F) 라고 하자. S와 Minimal IR을 합친 전체시스템의 최대 지연시간은 시스템 S만의 최대 지연시간과 동일하다. 즉 모든 플로우 f 와 $n, n > 0$,에 대해서 $\sup_{f,n}(D_n^f - A_n^f) = \sup_{f,n}(E_n^f - A_n^f)$.

Minimal IR은 “플로우 FIFO 시스템” S의 최대 지연시간을 증가시키지 않는다. 여기서 플로우 FIFO 시스템이란 개별 플로우 f 에 대해서 FIFO를 보장해 주는 시스템을 말한다. A_n^f 는 플로우 f 의 n 번째 패킷의 입력시각, D_n^f 는 시스템 S에서의 출력시각, E_n^f 은 IR에서의 출력시각이다. 그림 3을 참고하라.

Proof: 이 증명은 [4]의 Theorem 4의 증명과 유사하다. 시스템 S의 최대 지연시간을 d 라 하자. “Damper”라는 가상 장치를 생각해보자. Damper는 시스템 S로부터 오는 모든 플로우 f 의 패킷을 정확히 $(A^f + d)$ 시점에 내보내는 가상의 장치이다. 특정 플로우 f 가 Π -regular 하다면 A^f 가 이미 arrival curve를 만족한다는 의미이고, 따라서 $(A^f + d)$ 의 출력 시점을 가지는 damper의 출력도 동일한 arrival curve를 만족한다. 즉 damper도 하나의 Π -regulator이다. (damper의 input은 혼합 플로우의 패킷 sequence (D, L, F) 이며 d 가 시스템 S의 최대 지연시간이므로 모든 n 에 대해서 $D_n \leq (A_n + d)$ 이다. damper의 출력 시점이 $(A_n + d)$ 이므로 damper는 해당 sequence에 대해서 FIFO이다.)

한편 minimal IR은, Theorem 2에 의해서, 다른 어떤 Π -regulator보다도 작거나 같은 delay를 보인다는

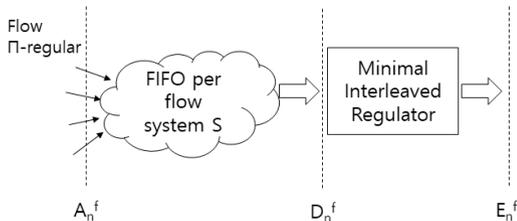


그림 3. 플로우 FIFO 시스템과 Minimal IR의 조합
Fig. 3. Combination of the flow FIFO system and the minimal interleaved regulator

것이다. 따라서 $E^f \leq A^f + d$. 즉, minimal IR의 출력시점은 $A^f + d$ 보다 항상 작거나 같다. 모든 플로우 f 에 대해서 이것이 성립하므로 Theorem이 증명되었다.

즉, Theorem 4에 의해서 FIFO 시스템이 각각의 플로우들에 대해서만 FIFO여도 기존 theorem이 성립한다. 작은 차이인 듯하지만 이러한 theorem의 확장으로 아래의 경우에도 IR의 사용과 theorem 적용이 가능하게 된다.

- 플로우 간의 priority 차이로 플로우 간 출력 순서가 바뀌는 경우
- 플로우 간의 경로 차이로 플로우 간 출력 순서가 바뀌는 경우

따라서 다양한 우선순위를 가진 플로우들이 인입되는 다양한 경로를 가진 하나의 네트워크에도 IR을 적용할 수 있다. 그림 3에서 f 가 관찰 대상인 플로우이며, 왼쪽의 시스템 S가 플로우 별로 FIFO를 보장해주는 네트워크이다. f 의 S에서의 최대지연시간을 D^f 라고 하자. S가 모든 플로우들에 대해서 최대지연시간 D 를 보장한다면, f 가 IR을 통과하는 시간을 포함한 최대지연시간도 D 가 되어 D^f 보다 커지게 된다. 대신 S에서 증가한 버스트 size가, 증가된 IR을 통과하면서 완화되어 원래 버스트 size 값으로 회귀하게 된다.

이러한 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 strict priority 방식으로 처리한다. 필요하다면 preemption도 적용할 수 있다.
- 높은 우선순위 플로우들은 아래와 같이 처리한다.
- 적당한 크기로 네트워크를 정의하여 네트워크의 output port 혹은 input port 마다 IR을 설치한다.
- 네트워크의 입력으로 Arrival curve를 따르는 최초 인입 플로우와 IR에서 출력되는 플로우만을 허용한다.
- 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 지연시간을 보장할 수 있으며, FA에 속한 플로우들의

최대지연시간도 보장된다. 네트워크 입력포트에서의 플로우 통합과 출력포트에서의 분리에 의해 버스트 크기가 증가하지만 IR을 통과하면서 원래 값을 되찾는다.

IV. 수치 분석

하나의 플로우 (혹은 통합 플로우) i 가 네트워크를 지나면서 LR 서버들만을 통과한다면, 이 플로우 i 의 패킷들이 겪는 단대단 지연시간의 최대치는 다음과 같은 식으로 표현된다.

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

아래에서 자주 사용하는 수학 기호에 대해서 표로 정리하였다.

(1)의 의미는 다음과 같다. 경로상에서 여러 개의 LR 서버를 지나더라도 플로우 지연시간의 최대치는 각각의 latency의 합에 최초 한 번의 최대 버스트에 의한 지연시간만을 더한 것이다. 이러한 LR 서버의 특성을 “Pay burst only once”라고 한다. 또한, LR 서버에 인입되는 플로우가 인입커브 (ρ_i, σ_i) 를 따른다면, LR서버를 통과한 후에는 인입커브 $(\rho_i, \sigma_i + \Theta_i^S \rho_i)$ 를 따른다. 즉 최대 버스트 크기가 $\rho_i \Theta_i^S$ 만큼 증가하게 된다^[8].

가장 이상적인 scheduler라고 할 수 있는 packetized generalized processor sharing (PGPS)과 간단한 라운드 로빈 기반의 deficit round robin(DRR)^[9]이 LR 서버에 포함된다. PGPS의 latency는 다음과 같다.

$$\Theta_i^{PGPS} = \frac{L_i}{\rho_i} + \frac{L_{\max}}{r}$$

퀀텀(quantum) 크기가 패킷의 최대길이보다 작은 경우를 포함하는, 일반적인 DRR의 latency는 다음과 같이 주어진다^[10].

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

여기서 F는 모든 플로우 ϕ_i 들의 합이다. F를 프레임의 크기라고도 한다. DRR의 동작에 대한 자세한 설명은 [9]를 참조하라.

본 장에서는 제안하는 프레임워크의 성능을 분석하고 더불어 플로우 기반 스케줄러를 사용하는 IntServ 프레임워크와의 비교를 위해 간단한 토폴로지를 제시한다. 제안하는 통합 플로우(FA) 기반 스케줄러와 IntServ의 스케줄러는 PGPS와 DRR의 두 가지 중 하나를 사용하도록 한다.

4.1 Symmetric 네트워크 환경

먼저 간단한 네트워크를 고려해보자. 그림 4와 같이 모든 플로우가 동일한 특성을 가지며, 동일하게 세 흐름을 거치는 네트워크이다. 모든 노드는 2개의 입력과 2개의 출력포트가 있다. 16개의 플로우가 입력포트로 인입되며 이 중 8개가 같은 포트출력된다. 두 번째 노드에서는 이 중 4개가 같은 포트출력되며 마지막 노드에서는 이 중 2개가 같은 포트출력된다. 따라서, 같은 {입력, 출력포트} 쌍을 가지는 플로우는 2개이며 통합 플로우 기반의 시스템에서 통합 플로우의 크기는 2이다. 이러한 입출력 패턴이 모든 노드에서 일어난다고 하자.

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
Θ_i^S	Latency of flow i at sever S_j
D_i	Delay experienced by packets of flow i

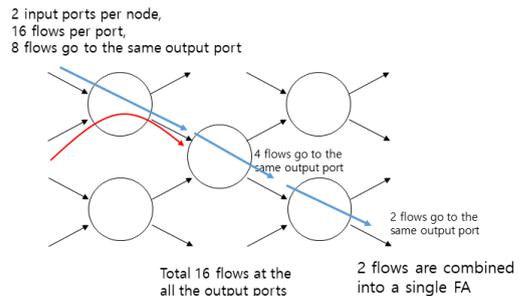


그림 4. 분석에 사용한 네트워크 구조
Fig. 4. Architecture of the single network used for the analysis

먼저, 개별 플로우 기반 스케줄링 방식에서의 지연 시간을 구하면, PGPS 스케줄러인 경우는

$$\Theta_i^{PGPS} = \frac{L_i}{\rho_i} + \frac{L_{\max}}{r} = \frac{L}{r/16} + \frac{L}{r} = \frac{17L}{r}. \quad \text{편의상}$$

$$\sigma_i = L \text{ 이라 하면 (1)에 의해서 } D_i \leq \sum_{j=1}^k \Theta_i^S = \frac{51L}{r}.$$

통합 플로우 기반 PGPS 스케줄링 방식에서는

$$\Theta_i^{PGPS} = \frac{L_i}{\rho_i} + \frac{L_{\max}}{r} = \frac{L}{r/8} + \frac{L}{r} = \frac{9L}{r} \quad \text{이며}$$

$$D_i \leq \sum_{j=1}^k \Theta_i^S = \frac{24L}{r}. \quad \text{따라서 1/2 이상 최대 지연시간을 줄이는 효과를 보인다.}$$

개별 플로우 기반 스케줄링 방식에서 DRR을 사용하는 경우를 고려해보자. 편의상 $\phi_i = L_i = L$ 이라고 하자. 단일 노드에서의 latency가

$$\begin{aligned} \Theta_i^{DRR} &= \frac{1}{r} \left[(F - \phi_i) \left(1 + \frac{L_i}{\phi_i}\right) + \sum_{n=1}^N L_n \right] \\ &= \frac{1}{r} \left[(16L - L) \left(1 + \frac{L}{L}\right) + \sum_{n=1}^{16} L \right] = \frac{46L}{r} \end{aligned}$$

이므로 전체 네트워크의 지연시간은

$$D_i \leq \sum_{j=1}^k \Theta_i^S = \frac{138L}{r}. \quad \text{통합 플로우 기반 DRR 스케}$$

줄링 방식에서는 $\Theta_i^{DRR} = 22L/r$, $D_i \leq 66L/r$. 따라서 이러한 symmetric 한 환경에서는 통합 플로우의 크기에 반비례하여 지연시간이 변화됨을 알 수 있다. Symmetric 한 환경에서는 더 나아가 이러한 네트워크가 연속되어도 성능 차이가 계속 선형적으로 벌어진다.

4.2 Asymmetric 네트워크 환경

이번에는 플로우 간의 특성이 달랐을 때 플로우의 지연시간이 어떻게 변화하는지 알아보자. 우리는 다른 플로우의 영향으로 관찰대상 플로우가 손해를 입는 경우를 분석하고자 한다. 플로우의 상대적인 전송속도 (rate)가 작을수록 네트워크에서의 지연시간이 커진다는 점을 활용한다. 따라서 그림 4의 관찰대상 플로우와, 해당 플로우와 통합되는 다른 하나의 플로우만 2배의 상대적인 전송속도를 할당받는다고 하자. 이 경우 개별 플로우 기반 스케줄링 방식에서의 지연시간을 구하면, $r = 9\rho_i$ 이며 PGPS 스케줄러인 경우는 $\Theta_i^{PGPS} = 10L/r$ 이며 $D_i \leq 30L/r$. 통합 플로우 기반 PGPS 스케줄링 방식에서는 $r = 9\rho_i/2$, $\sigma_i = 2L$, $\Theta_i^{PGPS} = 11L/2r$ 이며

$$D_i \leq \frac{\sigma_i - L}{\rho_i} + \sum_{j=1}^k \Theta_i^S = \frac{21L}{r}. \quad \text{따라서 FA 기반 스}$$

케줄링이 1.5배 가량 우수한 성능을 보인다. 개별 플로우 기반 스케줄링 방식에서 DRR을 사용하는 경우를 고려해보자. 편의상 $\phi_i = 2L_i = 2L$ 이라고 하자. 다른 하나의 플로우의 쿼터도 $2L$ 이며 이 두 플로우를 제외하면 모두 L 을 쿼터 값으로 가진다. 따라서 $F=18L$ 이다. 단일 노드에서의 latency가 $\Theta_i^{DRR} = 40L/r$ 이므로 전체 네트워크의 지연시간은 $D_i \leq 120L/r$. 통합 플로우 기반 DRR 스케줄링 방식에서도 관찰 대상 플로우가 속한 FA의 쿼터 값을 다른 FA의 2배로 정한다. $\Theta_i^{DRR} = 29L/2r$, $D_i \leq 48L/r$. DRR을 사용하는 경우도 FA 시스템의 성능이 상당히 우수하였다. 두 가지 스케줄러 모두 이러한 asymmetric 한 환경에서도 통합 플로우의 크기에 반비례하여 지연시간이 변화됨을 알 수 있다.

Asymmetric 한 환경에서는 다른 플로우의 최대 지연시간이 Minimal IR을 통과하면서 관찰대상 플로우에 영향을 주므로 전체 단대단 지연시간이 하나의 네트워크 지연시간과 다른 양상을 보일 수 있다. 그림 5와 같은 multi-domain 네트워크를 고려한다.

지금까지 고려했던 네트워크들이 연결되어 관찰대상 플로우가 세 개의 네트워크를 지난다고 가정하자. 모든 네트워크는 동일한 특성을 가지고 있다. Minimal IR을 지나면서 관찰대상 플로우의 최대 지연시간이 네트워크의 지연시간 최대치로 커진다. 따라서 asymmetric 환경에서 $\rho_j = r/18$ 인 플로우 j의 지연시간 최대치를 찾아보자.

개별 플로우 기반 스케줄링 방식에서의 i의 최대 지연시간은 변함없으며 통합 플로우 기반 스케줄링 방식에서의 최대 지연시간을 구하면, PGPS 스케줄러인 경우는 $\Theta_j^{PGPS} = \Theta_{\max}^{PGPS} = 10L/r$ 이며 $D_j = D_{\max} \leq 39L/r$. 따라서 세 개 네트워크를 통과

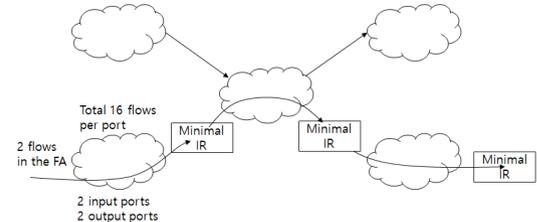


그림 5. 복수의 네트워크를 포함한 전체 네트워크 구조
Fig. 5. Architecture of the multiple networks used for the analysis

표 2. 지연시간 최대치의 비교
Table 2. Maximum delay comparison

	Flow-based system	FA-based system
PGPS, Single network	51L/r	24L/r
DRR, Single network	138L/r	66L/r
PGPS, Single network, Asymmetric flows	30L/r	21L/r
PGPS, Single network, Asymmetric flows	120L/r	48L/r
PGPS, Three networks, Asymmetric flows	90L/r	117L/r
DRR, Three networks, Asymmetric flows	360L/r	243L/r

하는 단대단 지연시간 최대치는 117L/r이다. 통합 플로우 기반 DRR 스케줄링 방식에서의 최대 지연시간을 구해보자. $\theta_{\max}^{DRR} = 24L/r$ 이며 $D_{\max} \leq 81L/r$. 따라서 단대단 지연시간 최대치는 243L/r 이다. 결과를 정리하면 아래와 같다.

V. 결 론

본 연구에서는 Interleaved regulator를 네트워크 사이에 배치하고 네트워크 안에서는 통합 플로우 기반 스케줄링을 함으로써 통해 플로우가 여러 네트워크를 지날 때도 낮은 복잡도로 지연시간의 최대치를 보장하는 방안을 제시하였다.

먼저 기존 연구들에서 제시된 Minimal interleaved regulator가 FIFO 시스템의 지연시간 최대치를 늘이지 않는다는 이론을 확장하여, FIFO 특성이 개별 플로우에만 적용되는 시스템에서도 해당 이론이 만족함을 증명하였다. 더 나아가 이러한 “플로우 FIFO 시스템”이 하나의 네트워크일 수 있다는 것을 제안하고 Minimal IR을 네트워크의 출력포트에 구현하며, 네트워크에서 플로우를 통합하여 스케줄러의 복잡도를 크게 낮추는 방안을 제시하였다. 구체적으로는 통합 플로우의 수가 네트워크의 (입력포트 수 x 출력포트 수)의 수준으로 결정되므로, 수백만 개에 이르는 플로우에 기반하는 스케줄러에 비해 100~1000배가량 복잡도가 낮아질 수 있다. 수치적 분석을 통해서 본 연구에서 제안한 낮은 복잡도의 통합 플로우 기반 시스템의 성능이 기존 integrated services (IntServ) 프레임워크에서 제시한 높은 복잡도의 개별 플로우 기반 시스템보다 비슷하거나 혹은 더 나은 수준의 성능을 보임을 확인하였다.

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] E. Mohammadpour, E. Stai, M. Mohiuddin, and J.-Y. Le Boudec, “Latency and backlog bounds in time-sensitive networking with credit based shapers and asynchronous traffic shaping,” *30th Int. Teletraffic Congress (ITC 30)*, 2018.
- [4] J.-Y. Le Boudec, “A theory of traffic regulators for deterministic networks with application to interleaved regulators,” *IEEE/ACM Trans. Netw.*, vol. 26, no. 6, Dec. 2018.
- [5] C.-S. Chang and Y. H. Lin, “A general framework for deterministic service guarantees in telecommunication networks with variable length packets,” in *Proc. 6th IWQoS*, pp. 49-58, May 1998.
- [6] J.-Y. Le Boudec, “Some properties of variable length packet shapers,” *IEEE/ACM Trans. Netw.*, vol. 10, no. 23, pp. 329-337, Jun. 2002.
- [7] 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.
- [8] D. Stiliadis and A. Varma, “Latency-rate servers: A general model for analysis of traffic scheduling algorithms,” *IEEE/ACM Trans. Netw.*, vol. 6, no. 5, Oct. 1998.
- [9] M. Shreedhar and G. Varghese, “Efficient fair queueing using deficit round-robin,” *IEEE/ACM Trans. Netw.*, vol. 4, no. 3, pp. 375-385, Jun. 1996.
- [10] L. Lenzi, E. Mingozzi, and G. Stea, “Tradeoffs between low complexity, low latency, and fairness with deficit round-robin schedulers,” *IEEE/ACM Trans. Netw.*, vol. 12, no. 4, Aug. 2004.

정진우 (Jinoo Joung)



1992년 2월: KAIST 전자공학과 학사

1994년 8월: NYU 전기전자공학과 Master

1997년 8월: NYU 전기전자공학과 Ph.D.

1997년 10월~2005년 2월: 삼성전자 종합기술원

2005년 3월~현재: 상명대학교 휴먼지능정보공학과 교수

<관심분야> 유무선통신, 네트워크, 임베디드 시스템
[ORCID:0000-0003-3053-9691]