

<https://doi.org/10.7236/IIBC.2019.19.1.63>

IIBC 2019-1-9

## Smoothing DRR: 스케줄링과 레귤레이션을 동시에 수행하는 서버

### Smoothing DRR: A fair scheduler and a regulator at the same time

정진우\*

Jinoo Jung\*

**요약** 최근 들어 스마트 팩토리, 차량 내 통신망, 광대역 전력망 등 다양한 응용분야에서 단대단 네트워크 지연시간(End-to-end network delay)에 대하여 엄격한 제한(bound)을 요구하고 있다. 이에 따라 기존 Integrated Services (IntServ) 프레임워크에서의 플로우 기반 스케줄러를 바탕으로 하는 해법들이 다시 제안되고 있다. 하지만 많은 수의 플로우를 구분하여 스케줄 하는 것은 복잡도가 커서 구현하기 어렵다. 차선책인 Class 기반 스케줄러는 노드를 지나면서 burst가 기하급수적으로 커지기 때문에 지연시간을 제한하기 어렵다. 이에 따라 트래픽 regulator와 class 기반 스케줄러를 앞뒤로 배치하여 burst의 크기를 줄이면서 동시에 스케줄러의 복잡도를 낮추려는 연구가 등장하고 있다. 본 연구에서는 이러한 regulator와 스케줄러가 하나의 서버에서 간단히 구현되는 방안을 제시하고 성능을 분석하여 대규모 네트워크에서도 수 ms 이내의 최대 delay를 보장함을 보였다.

**Abstract** Emerging applications such as Smart factory, in-car network, wide area power network require strict bounds on the end-to-end network delays. Flow-based scheduler in traditional Integrated Services (IntServ) architecture could be possible solution, yet its complexity prohibits practical implementation. Sub-optimal class-based scheduler cannot provide guaranteed delay since the burst increases rapidly as nodes are passed by. Therefore a leaky-bucket type regulator placed next to the scheduler is being considered widely. This paper proposes a simple server that achieves both fair scheduling and traffic regulation at the same time. The performance of the proposed server is investigated, and it is shown that a few msec delay bound can be achieved even in large scale networks.

**Key Words** : Scheduler, Regulator, Flow-based, Class-based, Delay bound

## 1. 서론

스마트 팩토리, 차량 간 통신, 차량 내 통신, 전문 음향 네트워크, 대규모 전력 제어망 등 다양한 응용분야에서 단대단 네트워크 지연시간(End-to-end network delay)에 대하여 수 msec에서 수초까지의 엄격한 제한(bound)

을 요구하고 있다<sup>[1,2,3]</sup>. 관련 국제 표준들도 속속 등장하고 있는데, IEEE 802.1 Time Sensitive Network (TSN)과 IETF Deterministic Network (DetNet)이 대표적인 표준이다. TSN은 IEEE 802.1의 TSN Task Group (TG)<sup>[4]</sup>에서 표준화를 진행하고 있는, 이더넷을 기반으로 지연시간(delay) 보장 및 무손실의 확정적 서비스를 제

\*정희원, 상명대학교 휴먼지능정보공학과  
접수일자 2018년 11월 30일, 수정완료 2019년 1월 5일  
게재확정일자 2019년 2월 8일

Received: 30 November, 2018 / Revised: 5 January, 2019 /

Accepted: 8 February, 2019

\*Corresponding Author: jjoung@smu.ac.kr

Dept of Human Intelligence and Information Engineering,  
Sangmyung Univ., Seoul, Korea

공하는 기술이다. IETF의 DetNet<sup>[5]</sup>은 TSN을 광역망으로 확장하려는 시도이다. 본 논문에서는 다음과 같은 서버를 제안하고 성능을 분석한다. 전체 트래픽은 high priority(HP)와 low priority(LP)로 나눈다. 스위치 등의 중계노드의 출력포트에서 HP 트래픽은 입력포트 별로 구분된다. 이렇게 구분된 여러 개의 HP 트래픽과 하나의 LP 트래픽을 각각 독립된 플로우로 본다. 즉 입력포트가  $N$ 개라면  $(N+1)$ 개의 플로우가 만들어진다. 이들 플로우를 공정하게 deficit round robin (DRR) 기반으로 스케줄해서 서비스한다. 다만 각각의 플로우를 저장하는 큐가 절대 비지 않도록 가상의 패킷을 생성해서 서비스한다. 결과적으로 간단한 입력포트 기반 비작업보존형 DRR 서버가 하나의 출력포트 당 하나씩 존재하게 된다. 본 논문은 다음과 같이 구성되었다. 다음 장에서 관련 표준화 및 연구동향을 살펴보고, 3장에서 상기한 서버의 구조를 제시하고 분석한다. 4장에서는 수치 분석을 통해 본 논문에서 제시하는 서버의 성능을 평가한다. 마지막으로 5장에서 본 논문에 대한 결론과 향후 방향을 제시한다.

## II. 관련 연구

### 1. IEEE 802.1 TSN 기술

IEEE 802.1 TSN TG에서 추구하는 기술적인 목표는 크게 Ethernet 프레임의 단대단 지연시간 보장 및 손실 가능성의 배제이다. 이를 위해 크게 포워딩 기술, 시간 동기화 기술, 경로 설정 및 자원 예약 기술, 신뢰성 확보 기술 등을 표준화하고 있다. 본 논문에서는 이 중 포워딩 기술을 연구의 대상으로 한다. 최근 새로운 방식의 포워딩 기술이 TSN TG에서 제시되었는데, Asynchronous Traffic Shaping(ATS)이 그 핵심이며 이 기술요소를 채용한 방식을 비동기식 방안이라고 부른다<sup>[6]</sup>. ATS는 Interleaved regulator라고도 하는데, 입력포트별, 클래스별 traffic regulation을 출력포트의 시작점에서 구현한다. 비동기식 방안은 ATS로 regulate된 트래픽에 정해진 시간구간 즉, 정해진 대역폭을 고정적으로 할당한다. 그러나 Class 별로 shaping을 수행한다고 하지만, ATS는 플로우별로 state를 저장하고 이를 바탕으로 입력포트별 클래스별 큐의 가장 앞에 위치하는 패킷이 속한 플로우의 상태정보에 따라 전송 시각을 결정해야 한다. 즉 큐는 하나로 유지하지만 플로우별 상태정보를 계속 갱신해야 하는 작업이 필요하며 따라서 상당한 복잡도를 내포한다.

본 연구에서는 플로우별로 관리해야 하는 상태정보가 배제된 입력포트 기반 scheduler를 제안한다.

### 2. IntServ 기반의 LR 서버 이론

90년대부터 활발히 연구되어 온 Integrated Services (IntServ)의 이론적 배경에 대해서 살펴보자. 다음의 세 가지 조건이 충족되면 패킷의 단대단 지연시간 최대치가 보장된다는 것이 IntServ의 핵심 이론이다. 1) 모든 인입 플로우의 {arrival rate, 최대 burst}는 일정 수준 이하로 규제되어, 소위 arrival curve에 합치한다(conform한다). 2) 모든 링크에서의 플로우들의 arrival rate의 합은 링크의 대역폭 (capacity) 보다 작다. 3) 모든 중계노드는 각각의 플로우에게 service를 제공하는데 이 때 일정 수준 이상의 service data rate를 정해진 latency (delay와는 다름) 이하의 시간부터 제공하여야 한다. 이 3)번 조건에서의 {service rate, latency}의 최소치를 만족하는 curve를 “service curve”라고 한다. 위의 조건 3)번의 service curve에 합치하는 service를 제공해 줄 수 있는 스케줄러를 LR 서버라고 한다<sup>[7]</sup>. LR서버는 다음과 같은 특징을 가진다.

**특징 1.** 연속한 LR 서버 두 개는 하나의 가상 LR 서버로 치환할 수 있는데 이 가상 LR 서버의 latency는 연속한 두 LR 서버의 latency들의 합과 같다.

**특징 2.** 하나의 플로우  $i$ 가 LR 서버들만을 통과 한다면, 이 플로우  $i$ 의 패킷들이 겪는 단대단 지연시간 (end-to-end delay)은 다음과 같은 식으로 표현된다.

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

여기서  $\rho_i$ ,  $\sigma_i$ ,  $L_i$ 는 플로우  $i$ 의 arrival rate과 최대 burst, 최대 패킷 길이이다.  $\Theta_j^S$ 는  $j$ 번째 LR 서버  $S_j$ 의 latency이다. 즉, 여러 개의 LR 서버를 지나더라도 각각의 latency의 합에 최초 한번의 max burst에 의한 지연시간 성분만을 더한 것이 지연시간 최대치이다. 이러한 특성을 “Pay burst only once”라고 한다.

**특징 3.** LR 서버에 인입되는 플로우가 arrival curve  $(\rho_i, \sigma_i)$ 를 따른다면, LR서버를 통과한 후에는 arrival curve  $(\rho_i, \sigma_i + \Theta_i^S \rho_i)$ 를 따른다. 이를 수식으로 표현하면,  $t_0$ 가 busy period의 시작 시점일 때,  $W_i(t_0, t) \leq \rho_i(t - t_0) + \phi_i + \rho_i \Theta_i^S$ 이다. 즉 max burst 크기가  $\rho_i \Theta_i^S$ 만큼 증가하게 된다<sup>[7]</sup>.

가장 이상적인 스케줄러인 GPS(Generalized Processor Sharing)에서의 패킷들의 서비스 완료 시점인 “Virtual finish time”을 계산해서 이것이 작은 패킷부터 순서대로 서비스 해주는 Packetized-GPS, Self-clocked fair queuing, Virtual Clock 등과, 간단한 round-robin 기반의 Deficit round robin(DRR)<sup>[8]</sup>과 Weighted round robin이 LR 서버에 포함된다. 이러한 LR 서버 중 virtual finish time 기반의 서버들은 일반적으로 N이 플로우의 수일 때 O(N) 혹은 O(logN)의 구현 복잡도를 보이기 때문에 코어 네트워크에서 구현되기 어렵다. 실제 Cisco와 Juniper의 코어 라우터에는 Round robin 기반 서버 특히 Deficit Round-Robin(DRR)이 구현되어 있다. DRR의 알고리즘은 다음과 같다<sup>[8]</sup>. 1) 각 큐들은 deficit값과 quantum 값을 할당 받는다. 2) Deficit의 초기 값은 0이고 quantum 값은 arrival rate과 비례하여 고정된 값으로 정해진다. 3) Backlog되어 있는 큐들은 매 turn마다 quantum 크기만큼 deficit값을 증가시킨다. 이 deficit값이 head of queue의 패킷 길이보다 크거나 같으면 해당 패킷을 서비스하고 deficit값을 그 패킷의 크기만큼 감소시킨다. 4) 큐가 비면 deficit값을 0으로 만든다. 5) 이러한 turn을 backlog되어 있는 큐들에 대해서 계속 반복한다.

Quantum 크기가 패킷의 최대길이보다 작은, 일반적인 DRR의 latency는 다음과 같이 주어진다<sup>[9]</sup>.

$$\Theta_i^{DRR} = \frac{1}{r} \left[ (F - \phi_i) \left( 1 + \frac{L_i}{\phi_i} \right) + \sum_{n=1}^N L_n \right]. \quad (2)$$

여기서  $\phi_i$ 는 플로우  $i$ 에 할당된 quantum 값이며  $F$ 는 모든 플로우들  $\phi_i$ 의 합이다.  $L_i$ 는 플로우  $i$ 의 최대 패킷 크기이다.

### III. 제안하는 서버의 구조

본 장에서는 LR 서버 이론을 바탕으로 입력포트 기반의 비작업보존형 LR 서버를 제안하고 이의 성능을 분석하도록 한다.

**Theorem 1.** 플로우  $i$ 가 시간구간  $(a,b]$  동안 계속 backlog되어 있다고 가정하자.  $(a,b]$  동안 DRR turn이 플로우  $i$ 에게  $k$ 번 서비스를 제공한다고 하자. 이 기간 동안 플로우  $i$ 가 받은 서비스의 총량  $W_i(a,b)$ 는 다음과 같이 제한된다.

$$k\phi_i - \delta_i^k \leq W_i(a,b) \leq k\phi_i + \delta_i^0 \quad (3)$$

여기서  $\delta_i^k$ 는  $i$ 의  $(a,b]$ 기간의 첫 번째 라운드부터 세어  $k$  번째 라운드의 끝 시점에서의 deficit값이다.

증명. [8]의 lemma 2의 증명과 동일함. ■

**Theorem 2.** Backlog 기간 중의 임의의 기간  $(a,b]$  동안 DRR 서버가 플로우  $i$ 에 제공하는 서비스의 총량은 다음과 같이 제한된다.

$$W_i(a,b) \leq \rho_i \frac{F_{\max}}{f} (b-a) + \phi_i + L_i. \quad (4)$$

여기서  $f$ 는 이 기간 동안 계속 backlog된 플로우들의 quantum값들의 총합이며  $F_{\max}$ 는 서버의 capacity와 플로우들의 arrival rate이 같은 가상의 상황에서의 quantum값들의 총합이다.

증명. 여기서의 backlog 기간이란 앞에서 정의한 “플로우 busy period”와는 다르며, 실제로 HP 트래픽의 패킷이 계속 큐에 있는 기간을 의미한다.  $a$ 에서 시작해서  $k$  번째 라운드가 끝나는 시점을  $t_k$ 라고 하자.  $t_0 = a$ 이다. 하나의 라운드 기간,  $(t_{k-1}, t_k]$ 의 길이

$$t_k - t_{k-1} = \frac{1}{r} \sum_{j \in B_k} (\phi_j + \delta_j^{k-1} - \delta_j^k) \text{이다. 여기서 } B_k \text{는}$$

이 기간  $(t_{k-1}, t_k]$  동안 backlog되어 있어 서비스를 받는 플로우들의 집합이다.  $(t_{k-1}, t_k]$  동안  $B_k$ 의 원소는 바뀌지 않는다. 여기서  $B$ 를  $(t_0, t_k]$  동안 계속 backlog되어 있는 플로우들의 집합으로 정의하자. 모든  $k$ 에 대해서  $B \subset B_k$ 이다. 여기서 다음의 부등식이 성립한다.

$$t_k - t_{k-1} \geq \frac{1}{r} \sum_{j \in B} (\phi_j + \delta_j^{k-1} - \delta_j^k). \text{ 이 부등식을 } k \text{에 대해서} \quad \text{모두} \quad \text{합하면}$$

$$t_k - t_0 \geq k \frac{f}{r} + \frac{1}{r} \sum_{j \in B} (\delta_j^0 - \delta_j^k) \geq \frac{f}{r} (k-1) \text{이다. 왜냐}$$

하면  $\sum_{j \in B} \delta_j^0 \geq 0$ 이고  $\sum_{j \in B} \delta_j^k \leq f$ 이기 때문이다. 따라서  $k \leq \frac{r}{f} (t_k - t_0) + 1$ 이다. Theorem 1로부터  $(t_0, t_k]$  기간 동안

$$\begin{aligned} W_i(t_0, t_k) &\leq k\phi_i + \delta_i^0 \leq \frac{r}{f} (t_k - t_0) \phi_i + \phi_i + \delta_i^0 \\ &\leq \rho_i \frac{F_{\max}}{f} (t_k - t_0) + \phi_i + L_i \end{aligned}$$

이다.  $L_i \leq \delta_i^0$ 이며  $\rho_i/r = \phi_i/F_{\max}$ 이기 때문이다. 즉,  $t_0$ 를 포함하여 언제나 deficit 값은 최대 패킷 길이  $L_i$ 보다 클 수 없다. 플로우의 arrival rate과 quantum값은 비례하

며,  $F_{\max}$  는 서버의 capacity와 플로우들의 arrival rate 이 같은 경우의 frame 크기이므로  $r$ 에 비례한다. 따라서  $t_k$ 와  $t_{k+1}$  사이의 임의의 시간  $b$ 에 대해서

$$\begin{aligned} W_i(a,b) &= W_i(a,t_k) \leq \rho_i \frac{F_{\max}}{f} (t_k - a) + \phi_i + L_i \\ &\leq \rho_i \frac{F_{\max}}{f} (b - a) + \phi_i + L_i \end{aligned}$$

이며 Theorem이 성립한다. ■

다음과 같은 algorithm으로 동작하는 비작업보존형 DRR서버를 생각해 보자.

1. 명시된 arrival rate 없이 인입하는 LP 트래픽 전체를 수용하는 플로우( $v$ )를 상정한다. 해당 플로우는 실제 HP 플로우들의 arrival rate의 총합과 서버의 capacity와의 차이만큼의 arrival rate을 가진 것처럼 취급된다 ( $\rho_v = r - \sum_i \rho_i$ ).
2. LP 플로우를 포함하여 모든 플로우의 큐가 항상 backlog 되도록, 큐의 서비스 차례 직전에 플로우의 큐가 비어 있으면 deficit 값을 0으로 만든 후 가상의 패킷을 생성한다. 생성되는 가상 패킷의 크기는 해당 플로우의 quantum 크기로 한다. 즉, 한번의 round에 서비스 될 수 있는 최대 크기의 패킷을 생성한다.
3. 가상의 패킷을 서비스하는 동안 해당 큐에 실제 패킷이 도착하면 그 즉시 가상 패킷의 서비스를 멈추고, deficit값을 0으로 만든 후 다음 차례의 큐를 서비스한다.
4. 가상 패킷의 서비스가 완료되어도 실제로 링크로 전송하지는 않는다.

위의 서버는 결과적으로 모든 플로우가 비작업보존 방식으로 처리된다. 이 서버를 Smoothing DRR (SDRR) 서버라고 정의한다. 비작업보존 스케줄러인 SDRR은 항상 모든 큐가 backlog 되어 있으며, 플로우들의 arrival rate의 총합이 서버의 capacity와 같다. 한편 Theorem 2에 따라, (a,b)에서 backlog된 플로우의 arrival rate의 총합이 서버의 capacity와 같은 경우, 플로우  $i$ 에 대한 서비스는 다음과 같이 제한된다.

$$W_i(a,b) \leq \rho_i(b-a) + \phi_i + L_i. \quad (5)$$

따라서 위에서 제안한 SDRR은 임의의 시간 구간에서 시간당 서비스 총량이 제한되어 bucket 크기가  $\phi_i + L_i$ 인 regulator를 통과한 효과를 가진다. SDRR 스케줄러를 통과한 입력포트별 통합플로우  $i$ 의 최대 burst는  $\phi_i + L_i$

이며,  $\phi_i$ 는 해당 통합플로우에 할당된 Quantum의 크기이다. 이렇게 입력포트별로 큐를 할당하여 SDRR을 적용하면 플로우별로 적용하는 것보다 complexity가 크게 줄어 결과적으로 quantum의 크기를 줄일 수 있다. 실제의 구현에서 적용 가능한 quantum의 크기는 수십 byte정도 일 것으로 예상된다.

이렇게 SDRR서버에서 입력포트 별 통합플로우  $i$ 의 패킷이 겪는 delay는 다음과 같다:

$$D_i \leq \frac{\sigma_i - L_i}{\rho_i} + \Theta_i^{SDRR}. \quad (6)$$

여기서  $\Theta_i^{SDRR}$ 는 SDRR 서버에서 통합플로우  $i$ 의 latency며 (3)으로 구한다.  $\sigma_i$ 와  $\rho_i$ 는 통합플로우의 max burst와 input rate이다. 위 시스템에서 SDRR의 output에서  $i$ 의 max burst는 Theorem 2과 LR서버의 특징 3에 의해서  $\phi_i + L_i$ 로 제한된다.

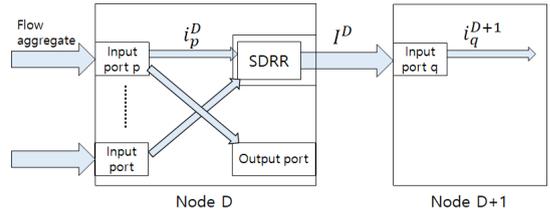


그림 1. SDRR 서버 구조와 통합플로우들의 Notation  
Fig. 1. SDRR server architecture and Notations for flow aggregates

위 그림에서 노드 D의 SDRR서버에서 출력된 HP 트래픽 전체를  $I^D$ 라고 하면,  $I^D$ 의 max burst  $\sigma_{I^D} = \sum \phi_i + L_i$ 이며  $I^D$ 가 전달된 다음 노드의 입력포트의 출력포트별 출력인 통합플로우  $i^{D+1}$ 의 max burst  $\sigma_{i^{D+1}}$ 는  $I^D$ 의 max burst  $\sigma_{I^D}$ 보다 같거나 작다. 왜냐하면  $A_{I^D}(t_0, t) = \sum A_{i^{D+1}}(t_0, t) = \sum (\sigma_{i^{D+1}} + \rho_{i^{D+1}}(t - t_0)) = \sum \sigma_{i^{D+1}} + \sum \rho_{i^{D+1}}(t - t_0) = \sigma_{I^D} + \rho_{I^D}(t - t_0)$ 이며 따라서 모든  $i^{D+1}$ 에 대해서  $\sigma_{i^{D+1}} \leq \sigma_{I^D}$ 이 성립한다.

## IV. Numerical Analysis

다음과 같은 토폴로지를 고려한다. HP에 속하는 플로우의 시작점과 도착점 사이에 6개의 bridge가 있다. 모든 bridge는 2개의 input과 2개의 출력포트가 있다. 관찰대

상인 플로우가 1번 입력포트로 들어와서 1번 출력포트로 나간다. 모든 bridge의 2번 입력포트로는 관찰 대상과 동일한 spec의 플로우가 들어와서 그 다음 노드의 2번 출력포트로 나간다. 그림에서 명시하지 않았지만 LP 트래픽도 존재한다.

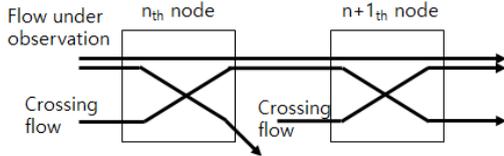


그림 2 수치 분석을 위한 서비스 시나리오  
 Fig. 2. Service scenario for Numerical analysis

아래는 분석에 사용한 parameter 값들이다.

표 1. 수치 분석에 사용된 파라미터  
 Table 1. Parameters used in Numerical analysis

Parameter	Value
L (Max packet length, both high & low priority traffic)	100~1500B
r (Link capacity)	100Mbps
$\sigma_i$ (Max burst size)	100~1500B
$\rho_i$ (Input data rate)	10~20Mbps
$\phi_i$ (Quantum size)	10~100B

노드에서의 최대 Delay는 SDRR에서의 최대 Delay ( $\frac{\sigma_i - L_i}{\rho_i} + \Theta_i^{SDRR}$ )로 구해진다. 첫 번째 노드를 제외한 이후 노드들에서의 최대 Delay는 input 플로우의 최대 burst 크기( $\sigma_i$ )가 변하는 점을 고려해야 한다. SDRR서버는 입력포트별 통합플로우의 max burst를  $\phi_i + L_i$ 로 제한한다. 따라서 해당 시나리오에서  $\sigma_i^p = 2(\phi_i + L_H)$ 이며 다음 노드의 SDRR서버로의 입력포트별 플로우의 max burst와 일치한다. 즉  $\sigma_{i+1} = 2(\phi_i + L_H)$ 이다. 이를 바탕으로 정리한 최대 delay는 다음과 같다.

먼저 아래 그림에서 각 플로우의 arrival rate을 10Mbps로, quantum 크기를 50Byte로 고정하고, 최대 패킷 길이를 100에서 1500Byte까지 변화시켰을 때의 최대 delay를 도시하였다.

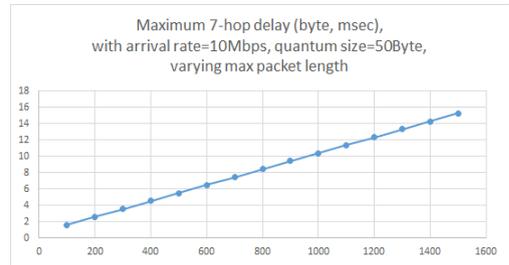


그림 3. 최대 패킷 크기 변화에 따른 최대 delay 변화  
 Fig. 3. Max delay with varying max packet length

그림 3에서와 같이 최대 패킷 길이가 100Byte일 때 6개의 bridge 노드를 통과한 패킷의 최대 delay는 1.59msec이다. 최대 패킷 길이가 1500byte일 때의 최대 delay는 15.3msec이다. 플로우의 arrival rate이 20Mbps로 좀 더 비중이 커지면 Frame의 크기가 작아지는 효과가 있으며, 아래와 같이 성능이 조금 향상된다.

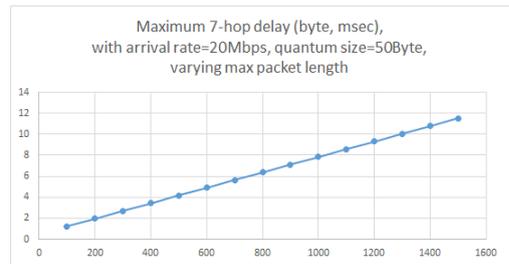


그림 4. 최대 패킷 크기 변화에 따른 최대 delay 변화  
 Fig. 4. Max delay with varying max packet length

이 경우 최대 패킷 길이가 100Byte일 때 6개의 bridge 노드를 통과한 패킷의 최대 delay는 1.23msec이다.

마지막으로 quantum size를 변화시키면서 최대 delay를 관찰하였다.

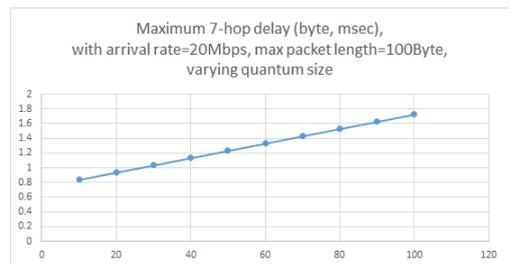


그림 5. Quantum 크기 변화에 따른 최대 delay 변화  
 Fig. 5. Max delay with varying quantum size

이 경우 quantum 크기가 10Byte에서 100Byte까지 큰 폭으로 변하는데 비해 최대 delay의 변화 폭은 크지 않았다. Quantum의 크기를 10Byte 까지 작게 해준 경우의 7 hop 최대 delay는 0.83ms 이다.

## V. 결론

본 연구에서는 입력포트 기반의 비작업보장형 서버로 scheduler와 regulator를 동시에 간단히 구현하는 방안을 제시하였다. 이를 통해 복잡한 대규모 네트워크에서의 단대단 delay를 보장하는 현실적인 해결책이 가능함을 증명하였다. 이러한 간단한 구조에도 불구하고 최대 패킷 길이와 quantum 크기를 적당하게 제한하면 7 hop에 네트워크에서 수 ms 이하의 최대 delay를 얻을 수 있었다. 향후 연구에서는 DRR보다 좀 더 복잡한 형태의 virtual finish time에 기반을 둔 LR서버를 비보장형으로 수정하여 적용하는 방안을 연구할 예정이다. 일반적인 LR서버의 regulation 기능 추가가 가능해 질 전망이다.

## References

- [1] M. Park, M.Yoon, K. Na, D. Kim, "A Study on Application of Time-Triggered Ethernet for Vehicle Network." Journal of the Institute of Internet, Broadcast and Communication, VOL. 15 NO. 6, December 2015, pp.79-88.  
DOI:10.7236/JIIBC.2015.15.6.79.
- [2] J. Lee, G. Lee, "An End-to-End QoS Control Method for Heterogeneous Networks," Journal of the Korea Academia-Industrial cooperation Society, Volume 10 Issue 10, Pages.2715-2720, 2009.  
DOI:10.5762/KAIS.2009.10.10.2715.
- [3] S. Oh, K. Jung, "A Reliable Protocol for Real-time Monitoring in Industrial Wireless Sensor Networks," Journal of Korea Institute of Information & Electronic Communication Tech. 2017, Vol.10 No.5  
DOI: http://dx.doi.org/10.17661/jkiect.2017.10.5.424
- [4] IEEE 802.1 Time-Sensitive Networking Task Group Home Page, <http://www.ieee802.org/1/pages/tsn.html>
- [5] IETF DetNet Working Group, <https://datatracker.ietf.org/wg/detnet/about/>
- [6] F. Dürr and N. G. Nayak. "No-wait Packet Scheduling for IEEE Time-sensitive Networks (TSN)." Proceedings of the 24th International Conference on Real-Time Networks and Systems. Pages 203-212, October 19 - 21, 2016  
DOI:10.1145/2997465.2997494
- [7] D. Stiliadis and A. 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/TNFCOM.1996.497884
- [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 (TON). Volume 12 Issue 4, August 2004 Pages 681-693  
DOI: 10.1109/TNET.2004.833131

### 저자 소개

#### 정진우(정회원)



- 1992년 : KAIST 전자공학과(학사)
- 1997년 : NYU School of Engineering (Ph.D in EE)
- 1997년 ~ 2005년 : 삼성종합기술원
- 2005년 ~ 현재 : 상명대학교 컴퓨터 과학과 교수
- 관심분야 : 유무선 네트워크, SoC design, Embedded system, 인공지능, 음성인식

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