The Journal of The Institute of Internet, Broadcasting and Communication (IIBC)
Vol. 18, No. 6, pp.121-126, Dec. 31, 2018. pISSN 2289-0238, eISSN 2289-0246

https://doi.org/10.7236/JIIBC.2018.18.6.121
JIIBC 2018-6-15

비작업보존 스케줄러를 갖는 IEEE 802.1 TSN에서 단대단 지연시간 보장

End-to-end Delay Guarantee in IEEE 802.1 TSN with Non-work conserving scheduler

정진우*

Jinoo Joung*

요 약 IEEE 802.1 TSN(Time Sensitive Network) TG(Task Group)는 이더넷을 기반으로 지연시간 보장 및 패킷 무손실 서비스를 제공하는 기술의 표준화를 진행 중이다. 본 연구는 다양한 TSN 기술 중 패킷 포워딩 기술에 주목한다. TSN의 포워딩 기술은 크게 동기형과 비동기형으로 구분할 수 있다. 동기형은 시간 동기화 기술을 바탕으로 정해진 시간 구간을 정해진 class에 할당하는 기술이지만 대규모 네트워크에서 사용하기 어렵다. 비동기 기술은 트래픽 regulation과 class 별 스케줄링을 바탕으로 지연시간 보장을 약속하지만 필요 이상으로 복잡한 구조를 가진다. 본 논문에서는 비동기형 TSN 네트워크 구조를 보다 간단히 만들면서도 지연시간을 보장하는 방안을 제시하였다. 이를 통해 플로우의 상태를 저장하여 regulation 결정에 사용해야 하는 복잡성을 배제할 수 있었다. 이러한 간단한 구조에도 불구하고 높은 우선순위 트래픽의 최대 패킷길이를 일정 수준 이하로 제한하면 TSN의 요구사항을 만족시킴을 보였다.

Abstract IEEE 802.1 TSN TG is developing standards for end-to-end delay bounds and zero packet loss based on Ethernet technology. We focus on packet forwarding techniques. TSN packet forwarding techniques can be classified into Synchronous and Asynchronous framework. Synchronous approach allocates fixed time period for a class, yet is complex for large networks. Asynchronous approach provides delay guarantee by regulator-scheduler pair, yet is unnecessarily complex, too. We propose network components for TSN Asynchronous architecture, which remove the complexity of maintaining flow state for regulation decisions. Despite such a simplicity, the proposed architecture satisfies the TSN's delay requirements provided the limited high priority traffic's maximum packet length.

Key Words : TSN, Ethernet, ATS, Delay bound

Ⅰ. 서 론

Time Sensitive Network (TSN) 기술은 IEEE 802.1의 TSN Task Group (TG)[1]에서 표준화를 진행하고 있는, 이더넷을 기반으로 지연시간(delay) 보장 및 무손실의 확정적 서비스를 제공하는 기술이다. TSN 기술은 오디오/비디오 전송을 목적으로 하는 Residential Ethernet 기술에서 출발해서 좀 더 전문적인 응용의 AVB(Audio Video Bridging) 기술로 확장되었다가 스마트 팩토리 등 산업용 네트워크까지 추가하여 다양한 실시간성 응용분야를 이더넷 기반의 범용 인터페이스로 통일하려는 시도이다. 본 논문에서는 현재 TSN에서 제시되는 단대단 지

*정회원, 상명대학교 휴먼지능정보공학과
접수일자: 2018년 10월 5일, 수정완료: 2018년 11월 5일
게재확정일자: 2018년 12월 7일

Received: 5 October, 2018 / Revised: 5 November, 2018 / Accepted: 7 December, 2018
*Jinoo Joung, jjoung@smu.ac.kr
Dept of Human Intelligence and Information Engineering, Sangmyung Univ., Seoul, Korea

연시간 보장에 관한 기술들을 소개하고, 이들의 대안으로 간단한 round-robin 기반의 비작업보존형 스케줄러를 포함한 네트워크 구조를 제시하고 이의 성능을 분석한다. 본 논문은 다음과 같이 구성되었다. 다음 장에서 TSN 관련 표준화 및 연구동향을 살펴보고, 3장에서 대안을 제시하고 분석한다. 4장에서는 numerical analysis를 통해 본 논문에서 제시하는 네트워크 구조의 성능을 평가한다. 마지막으로 5장에서 본 논문에 대한 결론을 제시한다.

II. 관련 연구

IEEE 8021. TSN TG에서 추구하는 기술적인 목표는 크게 Ethernet 프레임의 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 기술에 중점을 둔다. TSN의 시작인 Residential Ethernet에서는 "7개의 hop에서 2ms의 최대 delay"라는 명확한 지연시간 최대치를 정의하였으며 이 기준이 현재까지 이어오고 있다[2]. 다만 Residential Ethernet과 이의 후속 기술인 AVB는 A/V 응용만을 고려하여 비교적 높은 대역폭을 가지는 소수의 플로우(flow; 같은 발신지/목적지, 같은 application을 가지는, 시간적으로 근접한 패킷들의 집합)가 기술 적용대상인데 반해서, TSN은 다수의 저 대역폭 산업용 sensor-actuator 네트워크에서의 플로우 까지 고려하고 있다는 것이 가장 큰 차이점이다[3].

TSN의 포워딩 기법들이 등장한 시기 순으로 정리해 보면 다음과 같다. 먼저 초기 AVB(2005년~2012년)에서는 Stream reservation protocol (SRP)과 Credit-based shaper (CBS)가 기존 802.1Q에서 정의된 class별 strict priority scheduling (SPS)과 같이 사용되도록 제시되었다. CBS는 1990년대부터 연구된 Integrated Services(IntServ) 프레임워크의 Token-bucket shaper와 유사하나 다만 적용 대상이 플로우가 아닌, 같은 priority를 가진 플로우들의 집합인 class라는 점이 다르다. 따라서 구현상의 복잡도가 훨씬 낮은 수준이다. AVB 표준에서는 SRP를 통해 약속된 {최대 Burst, 평균 data rate}의 두 가지 parameter가 각 stream별로 지켜지면, Class 단위로 shaping하는 것만으로 단대단 지연시간이 보장된다고 보았다.

하지만 2012년 이후 진행된 표준 활동에서 low-priority traffic에 대해서 preemption이 가능하도록, 그리고 처음부터 low priority 프레임의 최대 크기에 맞추어 guard-band를 설정할 수 있도록 하였다. 여기에 더해서 네트워크 전역에서 동기화된 clock을 이용하여 시간을 일정 구간으로 나누어 모든 노드가 해당 구간(slot)의 정확한 시작시점과 종료시점을 공유한 상태에서 특정 프레임이 특정 구간에서 특정 브리지에 의해 서비스 되도록 규정하였다. 이를 Cyclic Queuing & Forwarding (CQF)라고 한다. 이러한 방식으로 프레임을 여러 브리지의 slot들에 배치하는 문제는 잘 알려진 job-shop scheduling 문제로 치환할 수 있다[4]. 이러한 scheduling은 그러나 NP-complete하여 heuristic과 적당한 optimization을 이용하여 풀어야 하는 복잡한 문제이다. [4]에서 제시한 scheduling 방법에 따르면 대략 200개 정도의 stream이 존재하는 경우 scheduling solution을 제시하는데 200초 정도가 소요되어 사용자가 감당할 수 있는 수준을 넘어선다.

이러한 동기식 방안에 대비해서 새로운 방식의 해법이 TSN TG에서 제시되었는데, Asynchronous Traffic Shaping(ATS)이 그 핵심이며 이 기술요소를 채용한 방식을 비동기식 방안이라고 부를 수 있다. 이 비동기식 방안은 기존 AVB의 SRP+CBS+SPS 해법에 ATS를 추가한 것이다. ATS는 Interleaved regulator라고도 하는데, Per-input port, per-class traffic shaping을 output port의 시작점에서 구현하는 것이다.

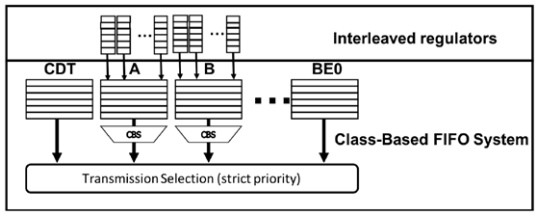

그림 1. ATS와 기존 Class 기반 FIFO 시스템
Fig. 1. ATS and Class based FIFO system

그러나 Class 별로 shaping을 수행한다고 하지만, ATS는 플로우 별로 state를 저장하고 이를 바탕으로 per-input port per-class queue의 가장 앞에 위치하는 패킷이 속한 플로우의 상태에 따라 전송 여부를 결정해야 하는 복잡성을 내포한다. 즉 queue는 하나로 유지하지만 플로우별 상태정보를 계속 갱신해야하기 때문에 만

만치 않은 작업이다. 본 연구에서는 플로우 별로 관리해야하는 상태정보가 배제된 진정한 per class 스케줄러를 제안한다.

Ⅲ. 제안하는 시스템 구조

90년대부터 활발히 연구되어 온 Integrated Services (IntServ)의 이론적 배경에 대해서 살펴보자. 다음의 세 가지 조건이 충족되면 패킷(혹은 프레임)의 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 Link에서의 플로우들의 arrival rate의 합은 link의 대역폭 (capacity) 보다 작다. 3) 모든 중계노드는 각각의 플로우에게 service를 제공하는데 이 때 일정 수준 이상의 service data rate를 정해진 latency 이하의 시간부터 제공하여야 한다. 위의 조건 3)번의 service curve에 합치하는 service를 제공해 줄 수 있는 스케줄러를 LR 서버라고 한다[5]. 가장 이상적인 GPS에서의 패킷들의 서비스 완료 시점인 "Virtual finish time"을 계산해서 이것이 작은 패킷부터 순서대로 서비스 해주는 PGPS, Self-clocked fair queuing, Virtual Clock 등과, 간단한 round-robin 기반의 Deficit round robin(DRR)[6]과 Weighted round robin이 LR 서버에 포함된다. 이러한 LR 서버들은 다음과 같은 특성을 가진다. 1) 연속한 LR 서버 두 개는 하나의 LR 서버로 치환할 수 있는데 이 때 하나의 LR 서버의 latency는 연속한 두 LR 서버의 latency들의 합과 같다. 2) 하나의 플로우 i가 LR 서버들만을 통과 한다면, 이 플로우 i가 겪는 단대단 지연시간(end-to-end delay)은 다음과 같은 식으로 표현된다.

$$D_i \le \frac{\sigma_i - L_i}{\rho_i} + \sum_{j=1}^{k} \Theta_j^{S_j} \qquad (1)$$

여기서 ρ_i와 σ_i는 플로우 i의 arrival rate과 max burst, $\Theta_j^{S_j}$ 는 j번째 LR 서버 Sj의 latency이다. 즉, 여러 개의 LR 서버를 지나더라도 각각의 latency의 합에 최초 한번의 max burst에 의한 지연시간만을 더한 것이 지연시간 최대치이다. 이러한 특성을 "Pay burst only once"라고 한다. 한편 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. Quantum 크기가 패킷의 최대길이보다 작은, 일반적인 DRR의 latency는 다음과 같이 주어진다[7].

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

여기서 ϕ_i는 플로우 i에 할당된 quantum 값이며 F는 모든 플로우들 ϕ의 합이다. L_i는 플로우 i의 최대 패킷 크기이다.

제안하는 스케줄러는 Interleaved regulator를, 단순한 input port별 트래픽을 하나의 플로우로 보고 이를 순서대로 처리하는 비작업보존형 DRR로 대체한다. 또한 Class based shaper(CBS)를 삭제한다. 비작업보존형 DRR은 다음과 같은 algorithm으로 동작한다.

1. 모든 플로우가 항상 backlog 되도록, queue가 비어있으면 가상의 패킷을 생성해서 서비스를 받도록 한다. 생성되는 가상 패킷의 크기는 quantum 크기로 한다. 2. 가상의 패킷을 서비스하는 동안 해당 플로우의 실제 패킷이 도착하면 그 즉시 가상 패킷의 서비스를 멈추고, deficit값을 0으로 만든 후 다음 큐를 서비스한다. 3. 가상 패킷의 서비스가 완료되어도 link로 전송되지는 않는다. 이 서버를 Smoothing DRR(SDRR)이라고 하자.

Theorem 1. 플로우 i가 시간구간 (a,b] 동안 계속 backlog되어 있다고 가정하자.(a,b] 동안 DRR turn이 플로우 i에게 k번 서비스를 제공한다고 하자. 이 기간 동안 플로우 i가 받은 서비스의 총량 Wi(a,b)는 다음과 같이 제한된다.

$$k\phi_i - \delta_i^k \le W_i(a,b) \le k\phi_i + \delta_i^0$$

여기서 δ_i^k는 i의 (a,b]기간의 첫 번째 라운드부터 세어 k번째 라운드의 끝 시점에서의 deficit값이다.

증명. [6]의 lemma 2의 증명과 동일함. ■

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

$$W_i(a,b) \le \rho_i \frac{F_{\max}}{f}(b-a) + \phi_i + L_i$$

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

quantum값들의 총합이다.

증명. a에서 시작해서 k번째 라운드가 끝나는 시점을 t_k라고 하자. $t_0 = a$이다. 하나의 라운드 기간, $(t_{k-1}, t_k]$의 길이 $t_k - t_{k-1} = \frac{1}{r}\sum_{j\in B_k}(\phi_j + \delta_j^{k-1} - \delta_j^k)$이다. 여기서 B_k는 이 기간 $(t_{k-1}, t_k]$ 동안 backlog되어 있어 서비스를 받는 플로우들의 집합이다. $(t_{k-1}, t_k]$동안 B_k의 원소는 바뀌지 않는다. 여기서 B를 $(t_0, t_k]$동안 계속 backlog되어 있는 플로우들의 집합으로 정의하자. 모든 k에 대해서 $B \subset B_k$이다. 여기서 다음의 부등식이 성립한다. $t_k - t_{k-1} \geq \frac{1}{r}\sum_{j\in B}(\phi_j + \delta_j^{k-1} - \delta_j^k)$. 이 부등식을 k에 대해서 모두 합하면 $t_k - t_0 \geq k\frac{f}{r} + \frac{1}{r}\sum_{j\in B}(\delta_j^0 - \delta_j^k) \geq \frac{f}{r}(k-1)$이다. 왜냐하면 $\Sigma_{j\in B}\delta_j^0 \geq 0$이고 $\Sigma_{j\in B}\delta_j^k \leq f$이기 때문이다. 따라서 $k \leq \frac{r}{f}(t_k - t_0) + 1$이다. Theorem 1로부터 $(t_0, t_k]$ 기간 동안

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

이다. $L_i \leq \delta_i^0$이며 $\rho_i / r = \phi_i / F_{\max}$이기 때문이다. 즉, t_0를 포함하여 언제나 deficit 값은 최대 packet 길이 L_i보다 클 수 없다. 플로우의 arrival rate과 quantum값은 비례하며, $F_{\max}$는 서버의 capacity와 플로우들의 arrival rate이 같은 경우의 frame 크기이므로 r에 비례한다. 따라서 t_k와 t_{k+1} 사이의 임의의 시간 b에 대해서

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

이며 Theorem이 성립한다. ■

Theorem에 따라 모든 플로우가 backlog되어있는 기간 (a,b]에서 플로우 i에 대한 서비스는 다음과 같이 제한된다.

$$W_i(a,b) \leq \rho_i(b-a) + \phi_i + L_i. \quad (3)$$

따라서 위에서 제안한 비작업보존 스케줄러인 Smoothing DRR은 시간당 서비스 총량이 제한되어 regulator를 통과한 효과를 가진다.

이러한 Smoothing DRR(SDRR)을 output port의 시작점에서 input port별로 통합된 플로우들에 적용하여 그림 1의 Interleaved regulator를 대체하는 것이 본 논문의 핵심 제안이다. SDRR을 input port별로 적용하면 플로우별로 적용하는 것보다 complexity가 크게 줄어 결과적으로 quantum의 크기를 줄일 수 있다. 실제의 구현에서 적용되는 quantum의 크기는 수십 byte정도일 것으로 예상한다. 한편 이렇게 regulate된 트래픽을 input으로 받는 Class-based FIFO 시스템에서, 최고 priority class에 속한 개별 플로우의 지연시간은 다음의 theorem으로 구할 수 있다.

Theorem 3. FIFO 혹은 Strict priority (SP) 서버는 개별 플로우에 대해서 LR 서버이며 플로우 i의 서버 S에서의 Latency Θ_i^S는 다음과 같이 주어진다: $\Theta_i^S = (\sigma^S - \sigma_i^S)/r^S + \Theta^S$. 이 때 σ^S는 모든 σ_i^S의 합이며, r^S는 서버 S의 link capacity이며, Θ^S는 FIFO의 경우 L/r^S, SP의 경우 $(L + L_i)/r^S$이다. L_i는 플로우 i의, L은 모든 트래픽의 최대 패킷 길이이다.

증명. [8]의 Theorem 1의 증명과 동일함. ■

이렇게 SDRR 서버와 SP 서버를 통과한 input port 별 통합플로우의 delay는 다음과 같다:

$$D_i \leq \frac{\sigma_i}{\rho_i} + \Theta_i^{S_{SDRR}} + \Theta_i^{S_{SP}}. \quad (4)$$

여기서 $\Theta_i^{S_{SDRR}}$과 $\Theta_i^{S_{SP}}$는 SDRR 서버와 SP 서버에서 통합플로우 i의 지연시간(latency)이다. σ_i와 ρ_i는 통합플로우의 max burst와 input rate이다. σ_i는 통합플로우가 통과한 이전 노드의 output port에서 방출되는 통합플로우의 max burst보다 같거나 작다.

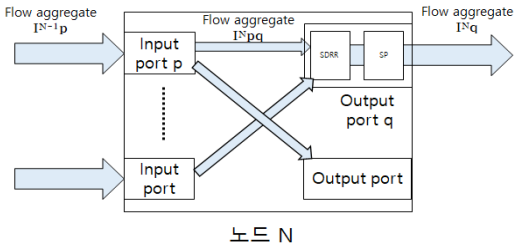

그림 2. 제안하는 시스템의 노드 구성도
Fig. 2. Proposed Node architecture

위 그림의 notation을 따르면, $\sigma_{I_{pq}^N} \leq \sigma_{I_p^{N-1}}$이다.

결과적으로 본 제안에서는, 단일 플로우에 대해서 LR 서버를 적용하지 않는 데에 따라 매 노드마다, 이전 노드에서 축적된 burst가 delay에 영향을 끼친다. 다만 이는 complexity를 고려하면 필연적인 선택이며, 이러한 영향은 SDRR의 regulation으로 최소화 된다. 특히, burst 크기가 latency에 영향을 미치고 이것이 다시 delay에 영향을 미쳐서 다음 노드의 burst를 증가시키는 feed-forwarding 효과의 고리를 regulation으로 끊는다는 것에 큰 의미가 있다.

Ⅳ. Numerical Analysis

다음과 같은 토폴로지를 고려한다. 플로우의 시작점과 도착점사이에 6개의 bridge가 있다. 모든 bridge는 2개의 input과 2개의 output port가 있다. 관찰대상인 플로우는 1번 input port로 들어와서 1번 output port로 나간다. 모든 bridge의 2번 input port로는 관찰 대상과 동일한 spec의 플로우가 들어와서 그 다음 노드의 2번 output port로 나간다. 낮은 priority의 트래픽도 존재한다. 따라서 high priority traffic은 전체 link capacity의 일부만 사용한다.

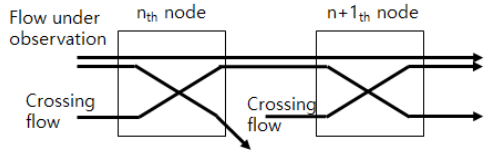

그림 3 수치 분석을 위한 서비스 시나리오
Fig. 3. Service scenario for Numerical analysis

아래는 분석에 사용한 parameter 값들이다.

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

Parameter	Value
L (Max packet length)	100~1500B
r (Link capacity)	100Mbps
σ_i (Max burst size)	100~1500B
ρ_i (Input data rate)	10Mbps
ϕ_i (Quantum size)	100B

노드에서의 Latency는 SDRR에서의 latency와 SP서버에서의 latency의 합으로 구해진다. 첫번째 노드를 제외한 이후 노드들에서의 latency는 input 플로우의 최대 burst 크기가 변하는 점을 고려해야 한다. 일반적으로 노드로 들어오는 플로우의 최대 burst 크기는 이전 노드에서의 latency에 input rate을 곱한 만큼 증가한다. 즉, $\sigma_i^{n+1} = \sigma_i^n + \rho_i \Theta_i^{S_n}$ 이다. 하지만 본 연구에서 주목할 점은 SDRR서버가 플로우의 max burst를 $\phi_i + L_i$로 줄인다는 것이다. 따라서 첫번째 노드를 포함한 모든 노드에서 SP서버로의 max burst는 플로우 당 $\phi_i + L_i$로 제한된다. SP 서버의 출력 지점에서의 max burst $\sigma_i^{n+1} = \phi_i + L_i + \rho_i \Theta_i^{S_n^{SP}}$ 이다. 하지만 이 burst가 다음 노드의 SDRR에서 다시 $\phi_i + L_i$로 줄어들게 된다.

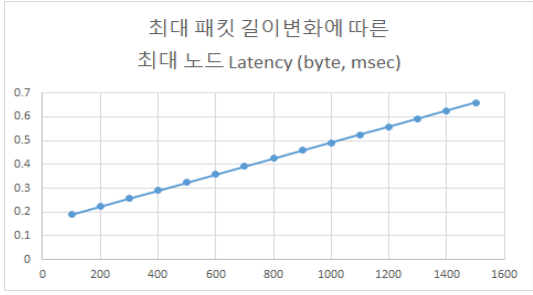

그림 4. 최대 패킷 길이 변화에 따른 최대 노드 latency
Fig. 4. Max nodal latency with variable max packet size

그림 4에서와 같이 High-priority traffic의 최대 패킷 길이가 1500Byte일 때 0.66msec의 노드 당 latency를 보인다. 최대 패킷 길이를 100byte로 제한하면 0.1869msec의 latency를 얻는다. 위와 같은 bridge 노드가 6개 연결되어 있는 토폴로지, 즉 7-hop 네트워크에서의 단대단 최대 delay는 다음 식으로 표현된다.

$$D_i \le \frac{\sigma_i}{\rho_i} + \sum_{n=1}^{6} (\Theta_i^{S_{SDRR}^n} + \Theta_i^{S_{SP}^n}) \,. \qquad (5)$$

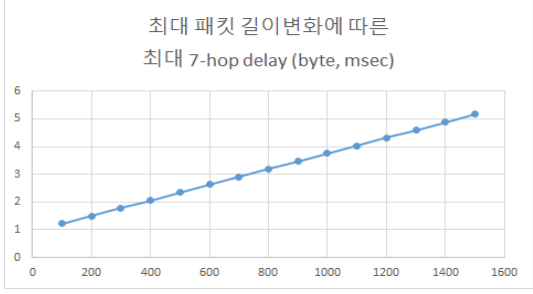

그림 5. 최대 패킷 길이 변화에 따른 최대 단대단 지연시간
Fig. 5. Max end-to-end delay with variable max packet size

그림 5에서 6개의 bridge 노드를 통과한 패킷의 최대 delay를 표시하였다. 최대 패킷 길이가 400byte 이하로 제한된다면 2msec의 요구조건을 만족하는 것을 확인할 수 있다.

V. 결 론

본 연구에서는 IEEE 802.1 TSN TG에서 표준화 진행 중인 보장성 네트워크 구조를 발전시키는 방안을 제시하였다. TSN의 비동기화 구조에서 ATS를 SDRR로 대체하고, CBS를 삭제하는 것이 제안의 핵심이다. 이를 통해 플로우의 상태를 저장하여 shaping 결정에 사용해야하는 복잡성을 배제할 수 있었다. 이러한 간단한 구조에도 불구하고 High-priority 트래픽의 최대 패킷 길이를 400byte 이하로 제한하면 "7 hop에서 2ms 이하"라는 TSN의 요구사항을 만족시킴을 보였다.

References

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

[2] Residential Ethernet Tutorial, www.ieee802.org/802_tutorials/05-March/tutorial_1_0305.pdf

[3] Mi-Ryong Park, et.al. "A Study on Application of Time-Triggered Ethernet for Vehicle Network," The Journal of The Institute of Internet, Broadcasting and Communication(JIIBC), VOL. 15 NO. 6, pp.79-88, December 2015
DOI : 10.7236/JIIBC.2015.15.6.79

[4] Frank Dürr and Naresh Ganesh 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

[5] 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/INFCOM.1996.497884

[6] 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

[7] Luciano Lenzini, Enzo Mingozzi, and Giovanni 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

[8] Jinoo Joung, Byeong-Seog Choe, Hongkyu Jeong, and Hyunsurk Ryu. "Effect of Flow Aggregation on the Maximum End-to-End Delay." Proceedings of High Performance Computing and Communications. Sep. 2006. also in LNCS, volume 4208
DOI: 10.1007/11847366_44

저자 소개

정 진 우(정회원)

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

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