

# IEEE 802.1 Qbv 시스템에서 Guard time 최소화를 위한 CPLEX 코드 개발 및 검증

정찬우(2017124192), 변개령(2022124074)

지도교수: 이동준

## 요 약

본 논문에서는 IEEE 802.1 Qbv Time aware shaper (TAS)의 실시간 데이터 전송 환경에서 낮은 우선순위 트래픽 전송에 따른 TAS 트래픽 전송 실패를 막기 위해 Guard Band를 추가하고 이로 인한 대역폭 손실을 최소화하기 위한 기법을 제안한다. 기존 TSN 표준인 IEEE 802.1Qbv, 802.1Qbu, 그리고 최신 연구 동향(패킷 크기 기반 PAS 알고리즘, No-wait 스케줄링)을 분석하고, 이를 기반으로 CPLEX OPL 코드로 구현한 MILP 모델을 소개한다. 모델은 플로우별 서브플로우 반복, wrap-around Delta, offset, no-wait 경로 등을 고려하여 guard overhead를 최소화하도록 설계되었다. 시뮬레이션 결과 guard 타임 축합을 최소화하는 해를 도출하였으며, 본 모델이 실제 네트워크 스케줄링의 실시간성과 효율성을 개선함을 입증하였다.

Keywords : TSN, CPLEX, guard band, IEEE 802.1 Qbv

## I. 서 론

### 1.1 연구의 배경

Time-Sensitive Networking(TSN)은 이더넷 기반 네트워크에서 정해진 시간에 트래픽을 전송하고 시간 동기화를 유지하여 지연 및 지터를 최소화하는 표준 기술로, 산업 자동화, 차량 통신, 철도, 항공 등 다양한 분야에서 실시간성과 신뢰성을 보장한다[2-4]. IEEE 802.1Qbv 표준은 Time-Aware Shaper(TAS) 메커니즘을 통해 스케줄드 트래픽(ST)의 정시성과 신뢰성을 확보하지만, 이 과정에서 설정되는 가드 타임(guard time)은 낮은 우선순위 트래픽을 차단하여 네트워크 자원의 비효율과 전송 지연을 초래한다[3][2]. 특히, 가드 타임은 최대 프레임 크기에 기반해 설정되므로 네트워크 대역폭 낭비가 불가피하다[4]. 이러한 문제를 해결하기 위해 프레임 프리엠션(IEEE 802.1Qbu)[3][2], 패킷 크기 인지 스케줄링(PAS)[1][3], No-wait 스케줄링 등 다양한 기술이 연구되고 있지만, 하드웨어 복잡성, 확장성 한계, 실시간성 확보의 어려움 등으로 여전히 개선의 여지가 존재한다[1][2]. 특히 최근에는 No-Wait 스케줄링과 PAS 알고리즘을 통한 가드 타임 최소화 기법이 활발히 연구되고 있으며[1][2][3], 이러한 기법들은 wrap-around delta, offset 등을 도입해 플로우 간 충돌을 방지하고 연속 전송을 가능하게 한다[2][3].

### 1.2 연구의 개요

본 논문에서는 TSN 스케줄링에서 가드 타임을 최소화하

고 네트워크 효율을 극대화할 수 있는 최적화 모델을 제안한다. 이를 위해 기존 표준(IEEE 802.1Qbv, Qbu)[2][3]과 최신 PAS[1][3], No-wait 스케줄링 기법의 한계와 문제점을 분석하고, CPLEX OPL을 기반으로 수학적 MILP 모델을 설계하였다[3][4][5]. 제안된 모델은 서브플로우 반복, wrap-around delta, offset, no-wait 경로를 고려해 guard overhead를 최소화하며, 임의의 데이터셋을 적용해 실제 네트워크 환경을 반영한 시뮬레이션을 수행하였다. 마지막으로, 실험적 결과와 기존 기법과의 비교 분석을 통해 guard time 최소화 및 네트워크 효율성 향상 효과를 검증하였다[3][5].

## II. 기존 연구 소개

### 2.1 TSN 및 가드 타임의 개념

TSN은 이더넷 네트워크의 시간 민감성을 개선하여 실시간성과 동기화를 보장하고, 네트워크 트래픽의 신뢰성을 확보한다[3][4]. 특히 IEEE 802.1Qbv 표준의 TAS(Time-Aware Shaper)는 포트 개폐를 통해 ST의 전송 시점을 제어하고, 이 과정에서 낮은 우선순위 트래픽의 간섭을 막기 위해 가드 타임을 설정한다[3][4]. 그러나 가드 타임은 포트 비활성화로 인해 대역폭 낭비 및 전송 지연을 유발하고, 네트워크 효율성을 저해하는 문제가 발생할 수 있다[3].

### 2.2 기존 Guard Time 최적화 기법

가드 타임 최소화를 위해 프레임 프리엠션(IEEE 802.1Qbu, 802.3br) 기술이 제안되어 왔다[3][2]. 이 기법

은 낮은 우선순위 트래픽 전송 도중 긴급한 고우선 트래픽이 도착하면 전송을 중단하고 우선적으로 전송한 후 나머지를 이어서 전송함으로써 자원 활용도를 높인다. 또한 PAS(Packet-size Aware Scheduling)는 가드 타임 동안 작은 패킷을 전송하여 대역폭 낭비를 줄이고 네트워크 효율을 개선하는 방식을 제안한다. 하지만 이러한 기법들은 하드웨어 구현 복잡성, 스케일링 문제, 실시간 환경 적용 한계 등으로 인해 여전히 한계가 존재한다.

### 2.3 No-Wait 스케줄링 연구

No-Wait 스케줄링(NW-PSP)은 가드 타임 없이 연속 전송을 가능하게 하는 대안으로, wrap-around delta, offset 등의 개념을 도입해 플로우 간 충돌을 방지하고[2][3] guard time을 최소화한다. 이 기법은 각 노드 간 스케줄을 최적화해 대기 시간을 제거하고, TSN 환경에서의 실시간성 및 네트워크 효율을 개선하는 효과를 기대할 수 있다. 본 연구에서는 이러한 No-Wait 스케줄링 개념을 MILP 모델로 수식화하고, CPLEX OPL 기반으로 구현해 기존 기법 대비 성능 개선 효과를 실험적으로 검증하였다[3][5].

## III. 시스템 모델 및 문제 정의

### 3.1 네트워크 모델

본 연구에서는 TSN 네트워크의 Unicast Scheduling을 대상으로 하여, no-wait 전송 제약, 프레임의 시작과 종료 시간 기반 guard 후보 선택 및 wrap-around guard 계산, offset 기반 시간 동기화 보정을 포함한 모델을 구축하였다. 네트워크는 N개의 노드와 E개의 링크, F개의 플로우로 구성되며, 각 플로우는 source, destination, period, duration 등의 속성을 가진다[2-4].

### 3.2 문제 정의 및 모델링

본 연구의 목적은 TSN(Time-Sensitive Networking) 네트워크의 스케줄링에서 guard time을 최소화하고, 스케줄 내 이벤트 간 간격을 줄여 스위치 간 동기화 정확도와 end-to-end delay를 최소화하는 것이다. 이를 달성하기 위해 CPLEX 기반의 MILP(Mixed Integer Linear Programming) 모델을 설계하고 최적화하였다. 목적 함수는 네트워크 내 각 링크별 guard time을 계산하여, 이들의 총합을 최소화하도록 설정하였다. 즉, 스케줄링 과

정에서 발생할 수 있는 guard band로 인한 대역폭 낭비를 최소화하고, 전송 지연 및 동기화 오차를 억제하는 방향으로 최적화를 수행한다. 모델은  $\Delta$  선택 변수와 offset 변수를 도입하여 각 플로우의 전송 간격을 정밀하게 조정하고, wrap-around를 적용하여 스케줄 간 충돌을 방지한다.  $\Delta$  변수는 각 링크의 후보 전송 시차(guard 후보값)를 나타내며, offset 변수는 전송 시점을 조정하여 각 플로우의 전송 타이밍을 동기화한다. 이를 통해 연속적인 no-wait 스케줄링을 달성할 수 있다. 또한,  $\Delta$  선택에 대해 one-hot 방식의 제약조건을 추가하여, 각 링크마다 하나의  $\Delta$  값만 선택되도록 설정함으로써 모델의 일관성과 최적화 안정성을 확보한다. 이로써 guard time 최소화 및 전송 경로 상 충돌 회피를 동시에 달성하며, TSN 네트워크의 스케줄링 효율성과 실시간성을 보장한다[3-5].

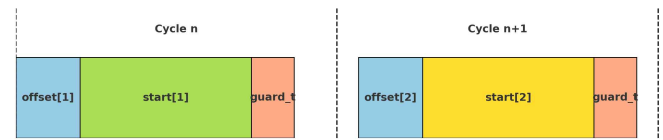


그림 1. IEEE 802.1Qbv 기반 스케줄 예시(모델 기반 표현)

그림 1은 IEEE 802.1Qbv에서 정의된 GCL(Gate Control List) 기반 Time-Aware Shaper(TAS) 구조를 본 연구의 스케줄링 변수들과 연계하여 표현한 것이다. TAS는 반복되는 주기(Cycle) 내에서 시간 슬롯별로 특정 큐의 게이트를 열고 닫는 방식으로 전송을 제어한다. 본 모델은 이러한 TAS 구조를 기반으로 offset[i], start[i], guard\_t 등의 변수들을 활용하여 각 플로우의 전송 시점을 정렬하고, 충돌을 회피하며 guard time을 최소화한다. 그림에서는 플로우의 시작 기준점을 나타내는 offset[i], 실제 전송 시점을 나타내는 start[i], 그리고 전송 간 보호 구간인 guard\_t가 시각적으로 표현되어 있으며, 이들은 MILP 기반 최적화 모델의 핵심 시간 변수로 활용된다.

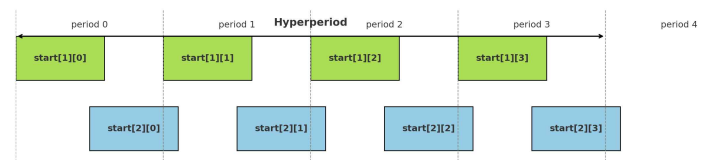


그림 2. Hyperperiod 예시

그림 2는 서로 다른 시작 오프셋과 주기를 가지는 두 개의 플로우가 hyperperiod 내에서 반복적으로 전송되는 스케줄 구조를 나타낸다. 각 전송 인스턴스는 주기 단위로 정렬되어 있으며, 플로우 간에는 연속적인 no-wait 스케줄링이 유지된다. 본 연구의 모델은  $\Delta$  후보값과 offset 변수를 조합하여 이러한 반복 구조를 자동으로 구성하고, 전송 간 guard time 이 최소화되도록 최적화한다. 그림은 이러한 구조가 hyperperiod 내에서 어떻게 정렬되는지를 시각적으로 보여주며, MILP 모델의 시간 정렬 메커니즘을 직관적으로 설명한다.

위의 목표를 달성하기 위해 다음과 같은 수식 및 제약조건을 설정하였다.

가. 목적 함수:

$$\min \sum_{f \in F} \sum_{e \in E} \sum_{sf \in SF_f} guard\_t_{f,e,sf} \quad (1)$$

나. 제약조건 정리:

본 연구에서 제안한 MILP기반 TAS 스케줄링 모델은 기존 논문의 No-Wait 및 연속 제약조건과 subflow 간 간섭 회피 조건[1][2]을 기반으로 하되,  $\Delta$  후보값 중 단일 선택을 강제하는 one-hot 제약(6), wrap-around 기반 guard 계산(7), 플로우의 물리적 경로 일관성을 보장하는 네트워크 연결성 제약(2)(3), 그리고 hyperperiod 내 시간 상한 제약(8) 등 기존 연구에 포함되지 않은 다수의 구조적 제약 조건을 추가하여 현실적이고 정밀한 스케줄링을 가능하게 한다.

(1) Offset 및 스케줄링 초기화

각 플로우의 offset은 1 이상, 해당 플로우의 주기 이하의 값을 갖는다. 또한 각 subflow(하나의 주기 내 또는 hyperperiod 내에서 실제로 전송되는 각 개별 인스턴스)의 시작 시간은 오프셋과 주기를 기준으로 정의된다.

$$1 \leq offset_f \leq period_f \quad (2)$$

$$startUsed_{f,e,sf} = offset_f + sf \cdot period_f \quad (3)$$

(단,  $e$ 는 source node의 out-edge)

(2) 플로우 경로 연결성 제약

각 플로우는 송신 노드, 중간 노드, 도착 노드로 구성되

며, 다음 조건을 만족해야 한다.

(송신 노드는 out-edge 1개, in-edge 없음)

$$\sum_{e \in E} outVE_{source(f),e} \cdot Use_{f,e} = 1 \quad (4)$$

$$\sum_{e \in E} inVE_{source(f),e} \cdot Use_{f,e} = 0 \quad (5)$$

(도착 노드는 in-edge 1개, out-edge 없음)

$$\sum_{e \in E} inVE_{dest(f),e} \cdot Use_{f,e} = 1 \quad (6)$$

$$\sum_{e \in E} outVE_{dest(f),e} \cdot Use_{f,e} = 0 \quad (7)$$

(중간 노드는 in-edge와 out-edge의 개수가 동일해야 함)

$$\sum_{e \in E} inVE_{v,e} \cdot Use_{f,e} = \sum_{e \in E} outVE_{v,e} \cdot Use_{f,e} \quad (8)$$

(3) 시간 변수 및 종료시간 계산

각 subflow의 종료 시간은 시작 시간에 duration을 더해 계산한다. 이때 해당 edge가 사용되는 경우에만 duration이 적용된다.

$$end\_t_{f,e,sf} = startUsed_{f,e,sf} + duration_f \cdot Use_{f,e} \quad (9)$$

(4) No-Wait 및 연속 전송 제약

인접한 edge 간에는 대기 시간 없이 연속 전송이 이루어져야 하며, 전달 시간 또한 이전 edge의 종료 시간과 일치해야 한다.

$$startUsed_{f,e_{out},sf} = startUsed_{f,e_{in},sf} + duration_f \quad (10)$$

$$tx\_eff_{f,e_{out},sf} = end\_t_{f,e_{in},sf} \quad (11)$$

(5) Subflow 간 충돌 방지 및 간격 유지

동일한 edge를 공유하는 subflow 간의 충돌을 방지하기 위해 전송 시간이 겹치지 않아야 하며, 같은 플로우 내의 subflow 간에는 고정된 주기 간격을 유지해야 한다.

$$startUsed_{f_1,e,sf_1} + duration_{f_1} \leq startUsed_{f_2,e,sf_2} \quad (12)$$

(if  $sf_1 \neq sf_2$ )

$$startUsed_{f,e,sf+1} - startUsed_{f,e,sf} = period_f \quad (13)$$

(6)  $\Delta$ (Delta) 계산 및 one-hot 선택

두 subflow 간의 시간 차이인  $\Delta$ 는 다음과 같이 정

의되며, 이 중 양수 값만 고려하여 하나만 선택되도록 one-hot 방식으로 표현한다. 또한 어떤  $\Delta$ 도 양수가 없는 경우에는 해당 subflow가 해당 edge에서의 첫 전송임을 의미한다.

$$d_{f, sf, fp, sfp, e} = startUsed_{f, e, sf} - end\_t_{fp, e, sfp} \quad (14)$$

$$\sum_{fp, sfp} isMin_{f, sf, fp, sfp, e} = hasPos_{f, sf, e} \quad (15)$$

$$isFirst_{f, sf, e} + hasPos_{f, sf, e} = 1 \quad (16)$$

(7) guard\_t 결정 로직

guard time은 전송 조건에 따라 다음과 같이 정의된다. 첫 전송인 경우 wrap-around 값을 적용하고, 그 외의 경우 가장 가까운  $\Delta$  값을 사용한다. 이후 cap 값에 따라 guard time의 최종 결정이 이루어진다.

$$isFirst = 1 \Rightarrow guardB = d + HP$$

$$isFirst = 0 \Rightarrow guardB = zSel \quad (17)$$

$$cap = 0 \Rightarrow guard\_t = guardB$$

$$cap = 1 \Rightarrow guard\_t = max\_guard\_time \quad (18)$$

(8) 시간 상한

모든 subflow의 시작 시간은 hyperperiod 내에 포함되어야 한다.

$$startUsed_{f, e, sf} \leq HP \quad (19)$$

## IV. 실험 및 결과 분석

### 4.1 실험 환경 및 시나리오

본 연구에는 Time-Sensitive Networking(TSN) 환경에서 guard time 최소화를 위한 스케줄링 최적화 모델의 성능을 평가하였다. 실험은 그림 1에서 볼 수 있듯이 노드 3개, 링크 3개(1-2-3)의 단순 네트워크에서 진행되었으며, 플로우 2개(모두 노드1→노드3)와 동일한 주기(1000 $\mu$ s) 및 전송시간(100 $\mu$ s)를 적용하였다. 하이퍼 피리어드는 데이터 확인을 위해 두 플로우의 주기 공배수인 3000 $\mu$ s로 설정하였다.

Network Topology: Node 1-2-3



그림3. 실험 네트워크 토폴로지

실험은 두 가지 조건으로 진행되었다. 첫째,  $\Delta$  선택 및 wrap-around 제약을 포함한 최적화 모델을 적용한 경우. 둘째, guard time을 고정(모든 구간에서 max guard time)으로 설정한 경우이다. 두 조건 모두 동일한 네트워크 및 입력 데이터를 사용하였으며, 최적화는 CPLEX OPL을 통해 수행되었다.

### 4.2 성능 지표 및 측정 방법

표 1. 변수 정의

목적 함수 값	모델의 최적화 목표 값
guard_t(가드타임)	플로우 별, 링크 별, 서브 플로우 별 guard time 값
offset	각 플로우의 시작 offset 값
startUsed	각 전송 구간의 시작 시각
Use	링크 사용 여부

guard\_t 값은 TSN 네트워크의 스케줄링 효율성을 직접적으로 나타내며,  $\Delta$  선택(one-hot)과 wrap-around 제약의 효과를 평가하는 주요 지표로 활용되었다.

### 4.3 결과 분석

#### $\Delta$ 선택 최적화 vs guard time 고정

본 연구에서는 TAS 환경에서  $\Delta$  선택 및 wrap-around 제약을 적용한 guard time 최소화 최적화 모델과 기존 guard time 고정 모델의 성능을 비교하였다. 비교 실험은 노드 3개(1-2-3)와 링크 3개로 구성된 단순 네트워크상에서 2개의 플로우를 대상으로 동일한 입력 데이터를 사용하여 수행하였다. 실험 결과는 목적 함수 값, guard time 설정 값, offset 조정, 그리고 네트워크 자원 효율성 측면에서의 성능 차이를 중심으로 분석하였다.  $\Delta$  선택

및 wrap-around 제약을 적용한 최적화 모델에서는 목적 함수 값이 600으로 guard time 고정 모델의 1200 대비 약 50% 감소하였다. 이는 guard time 고정 모델 대비 네트워크 효율성이 크게 향상되었음을 의미한다. 구체적으로,  $\Delta$  선택 모델은 플로우1의 모든 링크(1-2, 2-3)에서 guard time을 0으로 최소화하였으며, 플로우2의 경우에도 일부 링크를 제외한 대부분 구간에서 guard time을 최소화하는 결과를 도출하였다. 이러한 guard time 최소화는 플로우 간 충돌 방지 및 스케줄 동기화를 위해 offset [101, 1]이 적용된 결과로, wrap-around 제약의 효과를 입증하였다. 반면, guard time 고정 모델은 모든 플로우 및 링크에서 guard time이 최댓값(100)으로 고정되어 네트워크 자원의 비효율적 사용이 발생하였으며, 목적 함수 값 또한 1200으로 나타나  $\Delta$  선택 모델 대비 낮은 효율성을 보였다. 이 모델에서는 offset [1, 1]이 적용되어 최소 offset만으로 설정되었으며, 이에 따라  $\Delta$  선택 및 wrap-around 제약 모델 대비 충돌 회피 및 동기화 보정의 효과가 제한적이었다.

표 2. guard time 결과값 비교

조건	목적 함수	guard_t	guard_t	guard_t
		(f1 - E1)	(f1 - E2)	(f1 - E3)
최적화 적용	600	0	0	0
guard time 고정	1200	300	300	0
조건	guard_t	guard_t	guard_t	
	(f2 - E1)	(f2 - E2)	(f2 - E3)	
최적화 적용	300	300	0	
guard time 고정	300	300	0	

표2는 두 모델 간 guard time 값 및 최적화 결과를 정량적으로 비교하여 제시한다. 표 1에서는  $\Delta$  선택 및 wrap-around 제약 적용 모델과 guard time 고정 모델의 guard time 값의 차이를 명확히 비교하였다.

이 결과를 통해  $\Delta$  선택 및 wrap-around 제약을 포함한 guard time 최소화 최적화 모델은 네트워크 자원 효율성 및 guard time 절감 측면에서 기존 guard time 고정 모델 대비 우수한 성능을 나타냄을 확인하였다. 다만, 본 연구는 단순 네트워크 및 제한된 플로우 수를 대상으로 수행

되었으므로, 향후 연구에서는  $\Delta$  후보 확장, 경로 선택 최적화, 다양한 네트워크 토폴로지 및 부하 조건을 고려한 실험을 통해 제안된 모델의 실용성을 검증할 필요가 있다.

## V. 결론 및 향후 연구

본 연구에서는 TAS 환경에서 guard time을 최소화하기 위한 스케줄링 최적화 모델을 제안하였다. 제안된 모델은  $\Delta$  선택 및 wrap-around 제약을 적용하여 플로우 간 충돌을 방지하고 스케줄 동기화를 효과적으로 수행하였으며, 기존 guard time 고정 모델 대비 네트워크 자원 효율성을 높이고 guard time을 대폭 줄이는 성과를 확인하였다.

비록 본 연구의 실험은 단순화된 네트워크와 제한된 플로우 수를 기반으로 진행되었으나, 본 결과를 통해 TAS 환경에서의 guard time 최적화 가능성을 제시하였다. 향후 연구에서는  $\Delta$  후보 확장, 경로 선택 최적화, 다양한 네트워크 토폴로지와 부하 조건을 고려한 성능 평가를 통해 모델의 실용성을 더욱 검증할 예정이다.

## 참고문헌

- [1] Zhang, Chuwen et al., "Packet-size Aware Scheduling Algorithms in Guard Band for Time-Sensitive Networking," Noname manuscript, Tsinghua University, Fudan University, and Lehigh University, 미발행 원고.
- [2] Dürr, Frank, and Nayak, Naresh Ganesh, "No-wait Packet Scheduling for IEEE Time-sensitive Networks (fTSN)," Proceedings of the 24th International Conference on Real-Time Networks and Systems (RTNS '16), pp. xx - xx, Brest, France, ACM, Oct. 2016. DOI: 10.475/1234.
- [3] Pan, Zexian, Deng, Qingxu, Feng, Zhiwei, Lin, Yuhan, and Xia, Changqing, "Optimized Scheduling for Guard Bands Reduction in Time-Sensitive Networking," IEEE HPCC/SmartCity/DSS, pp. xx - xx, D O I : 10.1109/HPCC-DSS-SmartCity-DependSys57074.2022.00293, Aug. 2022.
- [4] Nayak, Naresh Ganesh, Dürr, Frank, and Rothermel, Kurt, "Exploring Practical Limitations of Joint Routing and Scheduling for TSN with ILP," Proceedings of the

25th International Conference on Real-Time Networks and Systems (RTNS '17), pp. xx - xx, ACM, Paris, France, Oct. 2017. DOI: 10.1145/3139258.3139289.

[5] Pahlevan, Maryam, Tabassam, Nadra, and Obermaisser, Roman, "How to Optimize Joint Routing and Scheduling Models for TSN Using Integer Linear Programming," IEEE Communications Standards Magazine, vol. X, no. Y, pp. xx - xx, Jan. 2019. DOI: 10.1109/MCOMSTD.2018.1700057.