

# 분산 실시간 시스템을 위한 양극단 여유도 기반의 우선순위 할당 방법

## End-to-End Laxity-based Priority Assignment for Distributed Real-Time Systems

김 형 육\*, 박 홍 성\*\*  
(Hyoung Yuk Kim and Hong Seong Park)

**Abstract** - Researches about scheduling distributed real-time systems have some weak points, not scheduling both sporadic and periodic tasks and messages or being unable to guaranteeing the end-to-end constraints due to omitting precedence relations between sporadic tasks. This paper describes the application model of sporadic tasks with precedence constraints in a distributed real-time system. It is shown that existing scheduling methods such as Rate Monotonic scheduling are not proper to be applied to the system having sporadic tasks with precedence constraints. So this paper proposes an end-to-end laxity-based priority assignment algorithm which considers the practical laxity of a task and allocates a proper priority to a task.

**Key Words** : End-to-end, Priority Assignment, Real-time scheduling

### 1. 서 론

분산 실시간 시스템은 응용 모델에 따라 비주기, 주기, 비실시간 태스크로 분류될 수 있는 다수의 태스크들과 비주기, 주기, 비실시간 메시지로 분류될 수 있는 다양한 메시지들로 이루어진다. 이렇게 다양한 태스크들과 메시지들로 구성되는 분산 실시간 시스템은 센서의 샘플링 시점에서부터 구동노드의 제어명령 출력까지의 양극단 시간제약, 어떤 태스크가 기동된 후에 어떤 메시지가 전송되고 다시 어떤 태스크가 기동되어야만 하는 태스크와 메시지 간의 선행제약, 두 개 이상의 센서노드로 구성되는 제어루프 경우 각 센서노드의 샘플링 시점 시간차를 어떤 범위 이내로 제한해야하는 센서 데이터 동기화 문제 등과 같은 양극단 제약사양이 존재하게 된다. 분산 제어시스템 스케줄링은 이러한 양극단 제약사양을 모두 만족하도록 태스크와 메시지에 적절한 우선순위와 주기를 설정하여야 한다. 일반적으로 실시간 시스템의 스케줄링은 NP-Hard 문제로 알려져 있기 때문에[1] 허리스틱(Heuristic) 방법을 사용하여 suboptimal 스케줄 결과를 얻는 방법들이 연구되어 왔다.

분산 실시간 시스템에서 양극단을 고려한 Schedulability 분석 연구로 어떠한 태스크도 두 개 이상의 메시지를 받지 않는다는 가정 하에서 생산자 태스크의 최악응답시간을 소비자 태스크 또는 메시지의 릴리즈 저터(Release Jitter)로 설정하는 방법을 이용하여 분산 실시간 시스템을 위한 Holistic Schedulability 분석 방법[2]이 제안되었고 태스크 기반의 스케줄링 방법[3]을 CAN으로 구성되는 분산 실시간 시스템으로 확장한 연구[4]에서는 여러 가지 소거법을 통하여 다중 루프를 구성하는 태스크와 메시지의 주기를 할당하여 시스템을 스케줄하는 방법을 제안하였다. 또한 [5]에서는 이진검색 알고리즘을 이용하여 다중 루프로 구성되는 분산 실시간 시스

템의 메시지 우선순위 설정 방법과 태스크 주기 할당 방법을 제안하였다. 그러나 이러한 연구들은 주기 태스크와 주기 메시지만을 고려한 것과 몇 가지 현실적이지 못한 가정 및 분산되어 있는 태스크들에 적절한 우선순위를 설정하는 방법을 제안하지 못했다는 단점을 가지고 있다. 비주기와 주기 태스크를 함께 고려하는 EDF나 LLF 계통의 on-line 스케줄 방법들은 태스크들에 우선순위를 설정하지만 비주기 태스크 간 선행제약을 고려하지 않았고 또한 분산 실시간 시스템과 같이 선행제약을 갖는 시스템에서 적절하지 않은 것으로 알려져 있다[6].

본 논문에서는 분산 실시간 시스템에서 비주기 태스크의 응용모델을 살펴보고 이를 간에 선행제약이 발생하는 경우 선행제약으로 인한 기존 우선순위 설정방법의 문제점을 서술한다. 또한 선행제약을 갖는 비주기, 주기 태스크와 비주기, 주기 메시지가 존재하는 분산 제어 시스템을 위한 양극단 여유도 기반의 우선순위 할당 방법을 제안한다.

### 2. 선행제약을 갖는 비주기 태스크

분산 실시간 시스템의 비주기 태스크 경우 어떤 노드 A에서 발생한 이벤트를 노드 A에서 로컬하게 처리하고 끝나는 형태의 태스크도 있지만 시스템 전체에 중대한 영향을 미칠 수 있는 알람 신호와 같은 이벤트는 로컬 노드의 비주기 태스크 뿐만이 아니라 시스템에 분산되어 있는 알람처리 태스크들에 의해 처리되어야 하며 이러한 태스크들 간에는 어떤 태스크가 다른 태스크를 기동시키는 선행제약이 존재하게 된다. 이러한 이벤트들은 시스템 전체에 영향을 미치기 때문에 발생한 이벤트에 대한 데드라인 즉, 처리 완료 시간 제약을 처음 기동되는 비주기 태스크의 기동시점에서 마지막으로 기동되는 비주기 태스크의 수행완료 시점까지로 보아야 한다. 예를 들어 발생한 알람에 대한 데드라인이 10ms이고 이 알람을 처리하는 비주기 태스크들이 센서노드의 태스크 A, 제어노드의 태스크 B, 구동노드의 태스크 C 가 선행제약을 가지며 순서대로 수행된다고 가정하자. 이 때 태스크 A, B, C가

#### 저자 소개

\* 김 형 육: 강원대학교 공과대학 전기전자정보통신공학부  
\*\* 박 홍 성: 강원대학교 공과대학 전기전자정보통신공학부

각각 기동된 시점에서 10ms의 데드라인을 만족하는지를 검사하여 schedulability 분석을 수행하면 안 되고 태스크 A가 기동된 시점-알람이 발생한 시점-에서부터 태스크 C가 완료되는 시점-알람에 대한 처리가 완료되는 시점-까지가 10ms를 만족하는지를 평가하여야 한다.

### 3. 실행제약으로 인한 Schedulability 영향

비주기 태스크와 주기 태스크가 존재하는 간단한 분산 실시간 시스템이 그림 1에 나타나 있다. 해당 시스템에 대한 요구사항 및 미리 주어지는 값으로서 각 태스크의 수행시간(*e*)과 데드라인(*d*)이 표 1의 하얀색 부분에 나타나 있다. 제어루프의 양극단 시간제약은 10ms이다. 분석의 편의성을 위해 메시지로 인한 지연은 없다고 가정하고 시스템의 각 노드가 고정 우선순위 기반의 선점형 스케줄러를 사용한다고 가정한다.

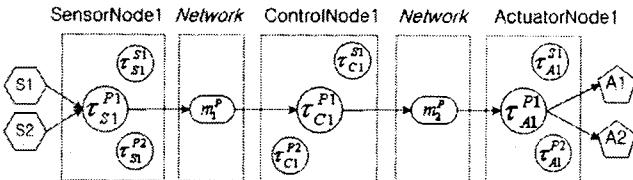


그림 1 비주기 태스크간 선행제약이 없는 경우

주어진 요구사항을 만족하면서 시스템이 운영될 수 있도록 태스크와 메시지에 적절한 주기와 우선순위를 부여하는 기존의 스케줄링 방법을 고려해 보자. 우선 태스크에 주기를 할당하는 방법으로 비주기 태스크의 경우 주기를 알 수 없으나 최대 기동빈도에 기반을 둔 가상주기 설정방법을 사용하고 제어루프의 주기태스크간 주기조화관계를 적용하고, 나머지 독립적인 주기태스크는 자신의 데드라인을 주기로 설정하면 표 1의 음영처리 부분의 주기(*T*)행과 같이 설정된다. 태스크 우선순위는 일반적으로 사용되는 RM 스케줄 방법[7]을 적용하면 태스크의 주기에 비례하게 표 1의 우선순위(*p*)행과 같이 설정된다.

표 1 RM을 이용한 그림1 시스템의 스케줄 결과

Task	$\tau_{S1}^{P1}$	$\tau_{S1}^{P2}$	$\tau_{S1}^{S1}$	$\tau_{C1}^{P1}$	$\tau_{C1}^{P2}$	$\tau_{C1}^{S1}$	$\tau_{A1}^{P1}$	$\tau_{A1}^{P2}$	$\tau_{A1}^{S1}$
<i>e</i>	2	3	2	2	2	4	2	3	2
<i>d</i>	10	10	15	10	10	15	10	10	15
<i>T</i>	10	10	15	10	10	15	10	10	15
<i>p</i>	1	2	3	1	2	3	1	2	3

비주기 태스크간 선행제약이 없는 그림 1의 시스템이 표 1과 같이 스케줄되어 운영된다면 이때의 태스크의 동작 타이밍도 그림 2에 나타나 있으며 제어루프의 양극단 데드라인 및 각 태스크의 데드라인이 만족되므로 시스템이 스케줄 가능함을 알 수 있다. 이때의 시스템 이용률은 SensorNode1-63.3%, ControlNode1-66.7%, ActuatorNode1-63.3%이다. 즉, RM을 이용한 스케줄 시 위와 같은 시스템 부하에서 시스템을 스케줄할 수 있음을 보이고 있다.

그러나 그림 3과 같이 비주기 태스크간 선행제약이 생긴다면 RM 방법은 같은 시스템 부하에서 시스템을 스케줄할 수 없게 된다. 그림 1의 시스템 모델에서 만약 시스템 전체에 영향을 미치는 알람 이벤트가 발생되어 시스템에 분산되어

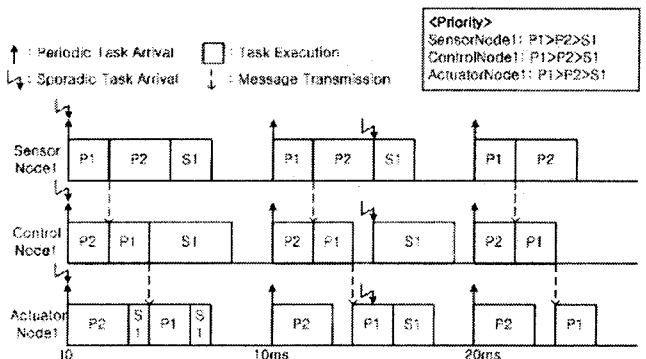


그림 2 표1에 따른 그림1 시스템의 태스크 수행도

있는 비주기 태스크들의 순차적인 수행을 통해 알람을 처리해야 된다면 이에 대한 시스템 모델이 그림 3에 나타나 있다. 그림 3의 시스템의 요구사항이 그림 1의 시스템과 같이 주어지고 해당 알람 이벤트 처리 태스크들로 이루어지는 이벤트 경로의 양극단 시간제약, 즉 알람 처리완료 시간이 15ms일 때 RM을 통한 스케줄 결과는 선행제약이 없는 시스템의 스케줄 결과인 표 1과 같게 된다. 이때 그림 4에 나타나 있는 것처럼 비주기 태스크간 선행제약으로 인해 이벤트 경로의 양극단 응답시간이 15ms의 데드라인을 만족하지 못하게 된다. 이때의 결과는 선행제약이 없을 때의 시스템 부하 조건에서 RM이 시스템을 스케줄할 수 있었지만 선행제약으로 인해 같은 시스템 부하에서도 RM이 시스템을 스케줄할 수 없음을 보이고 있다. 즉, 선행제약을 고려하지 않고 비주기 태스크를 스케줄할 경우 선행제약으로 인해 시스템의 schedulability가 보장되지 못할 수 도 있다.

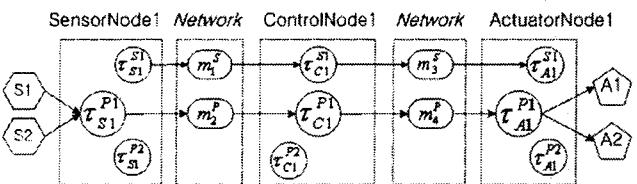


그림 3 비주기 태스크간 선행제약이 있는 경우

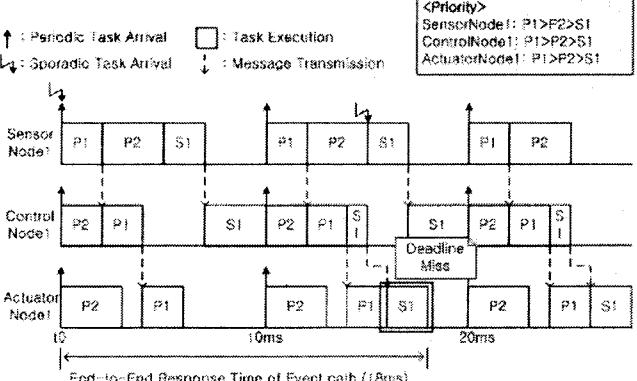


그림 4 표1에 따른 그림3 시스템의 태스크 수행도

### 4. 양극단 여유도 기반의 우선순위 할당

그림 3의 시스템 이용률을 감소시키지 않고 시스템을 스케줄 가능하게 하기 위해서는 좀 더 적절한 우선순위 할당방법이 필요하다. 각 노드에서 독립적으로 수행되는 주기태스크

들의 경우 데드라인이 10ms이지만 태스크 기동에서 다음 기동까지의 시간에서 수행시간을 뺀 여유시간은 각각 7ms 또는 8ms가 된다. 반면 이벤트 경로의 비주기 태스크들의 기동 주기는 15ms이고 여기서 수행시간을 빼면 위의 주기 태스크 보다 큰 여유도를 갖게 되지만 이 주기 이내에 세 개의 비주기 태스크가 완료되어야 하므로 이벤트 발생 주기에서 각 비주기 태스크의 수행시간을 뺀 여유도를 세 개의 태스크에 적절하게 매핑시켜야 비주기 태스크의 여유도를 고려해야 한다. 즉, 기동에서 데드라인까지의 여유도를 실제적으로 평가하여 우선순위를 설정해야 한다.

적절한 우선순위 할당을 위하여 태스크의 수행 여유도를 제어루프 또는 이벤트 경로의 양극단에서 분석하여 태스크의 우선순위를 할당하는 알고리즘이 그림 5에 나타나 있다. 제어루프나 이벤트 경로에 존재하는 태스크 경우 양극단 시간제약에서 해당하는 모든 태스크의 수행시간과 메시지 전송시간을 빼고 다시 태스크와 메시지의 수 만큼 나누어 각 태스크와 메시지의 여유도를 계산한다. 독립적으로 수행되는 태스크들은 자신의 데드라인에서 수행시간을 빼여 여유도를 계산한다. 모든 태스크와 메시지에 여유도가 계산되면 노드별로 또는 네트워크 상에서 여유도가 작은 태스크와 메시지 순으로 우선순위를 높게 부여하므로써 높은 이용률에서도 시스템을 스케줄할 수 있는 적절한 우선순위 할당이 이루어진다. 이 방법을 통해 그림 3의 시스템을 이용률 감소 없이 스케줄이 가능함이 그림 6에 나타나 있다.

```

/* Φ : the task set including all tasks in the system */
/* Ej : the set including all tasks and messages in the j-th event path */
/* Xj : the set including all tasks and message in the j-th control loop */
/* Ψk : the task set including all tasks in the k-th node */
/* laxi : the laxity of τi */
/* N(x) : the number of all tasks and messages included in a set x */
/* EventPathID(x) : the event path's index including a task x */
/* ControlLoopID(x) : the control loop's index including a task x */

foreach ∀ τi ∈ Φ do
    if τi is included in a control loop or a event path
        then if τi is a sporadic task then j := EventPathID(τi)
            laxi :=  $\left( MADT_j^{B_{max}} - \sum_{v_{t_i} \in E_j} e_i - \sum_{v_{m_i} \in E_j} C_i \right) / N(E_j)$ 
        else j := ControlLoopID(τi)
            laxi :=  $\left( MADT_j^{C_{min}} - \sum_{v_{t_i} \in X_j} e_i - \sum_{v_{m_i} \in X_j} C_i \right) / N(X_j)$ 
        else laxi := Ti - ei
    end
    foreach ∀ k do
        Set the priority of ∀ τi ∈ Ψk according to laxi
    end

```

그림 5 Laxity 기반 태스크 우선순위 설정 알고리즘

## 5. 결 론

분산 실시간 시스템에는 비주기 태스크, 주기 태스크, 비실시간 태스크와 같이 다양한 태스크들과 이러한 태스크간 대이터 교환을 위한 비주기, 주기, 비실시간 메시지들이 존재하

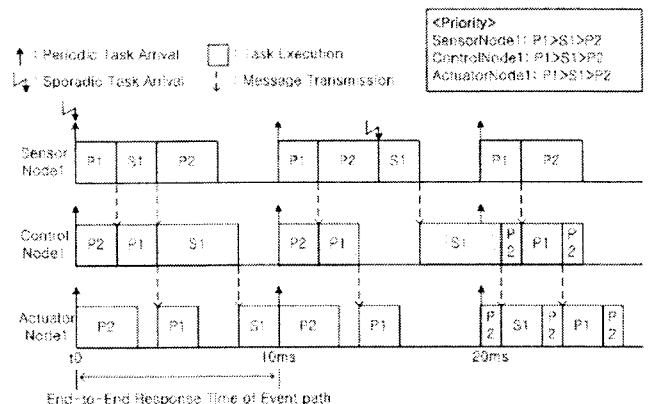


그림 6 우선순위 조절 후 그림3 시스템의 태스크 수행도

게 되므로 시스템의 요구사항을 만족하도록 이러한 태스크들과 메시지들을 스케줄하는 것은 매우 어려운 문제이다.

본 논문에서는 선행제약을 갖는 비주기 태스크 모델 및 로컬 노드 관점에서만 분석되던 비주기 태스크의 Schedulability 분석의 문제점을 서술하였다. 또한 양극단 제약을 고려하여 보다 높은 이용률 조건에서도 시스템을 스케줄 가능하게 하는 양극단 여유도 기반 우선순위 할당 방법을 제안하였다. 앞으로 연구할 과제로는 제안된 알고리즘을 좀 더 복잡한 모델에 적용해 보는 것과 on-line 스케줄링방법으로 적용하는 것이 있다.

## 참 고 문 헌

- [1] J. Y.-T Leung and J. Whitehead, "On Complexity of Fixed-Priority Scheduling of Periodic Real-Time Tasks," Performance Evaluation, 2(4), pp. 237-250, December, 1982.
- [2] K. Tindell and J. Clark, "Holistic schedulability analysis for distributed hard real-time systems", Microprocessing and Microprogramming, Vol.40, No.2, pp.117-134, Apr., 1994.
- [3] R. Gerber and S.S. Hong, "Guaranteeing Real-Time Requirements with Resource-Based Calibration of Periodic Processes," IEEE Tr. on Software Engineering, 21(7), July, 1995.
- [4] J.W. Park, Y.S. Kim, S.S. Hong, M. Saksena, S.H. Noh and W.H. Kwon, "Network conscious of Distributed Real-Time Systems," Journal of System Architecture, pp. 131-156, 1998.
- [5] Hyoung Yuk Kim and Hong Seong Park, Period and Priority Assignment Method for DCS Design, Asian Journal of Control, Vol.5, No.3, pp.422-432, Sept., 2003.
- [6] Giorgio C. Buttazzo, "Hard Real-Time Computing Systems," Kluwer Academic Publishers, pp.82-92, 1997.
- [7] C. L. Liu and J. W. Layland, "Scheduling algorithms for multiprogramming in a hard-real-time environment," Journal of the ACM, Vol.20, No.1, pp.46-61, 1973.