실시간 처리를 위한 IEEE 802.4 토큰버스 네트워크의 타이머 할당과 유용도 처리 성능 해석

김 정 호† 이 민 남†† 이 상 범†††

요 약

IEEE 802.4 토큰버스 방식이 산업용 네트워크에 있어서 표준으로 넓게 받아들여지고 있다. 802.4에서는 정보의 교환을 제어하기 위해 일종의 타이머를 사용하여 트래픽의 다중 클래스를 지원한다. 실시간 처리를 위한 802.4 기능의 시간할당방법은 실제 생산 공정에서 요구되는 실시간 제어응용에서 최악의 상태 억세스 요구를 만족하도록 타이머들이 설정되어야 한다. 물론 시간에 제약을 받지 않은 다른 응용들도 동시에 지원될 수 있다. 본 논문에서는 실시간 처리를 위하여 토큰 버스 네트워크 운영시 타이머 할당 구조에 따른 최소한의 대역폭 할당을 위하여 알고리즘을 제안하고 해석적 방법과 이의 시뮬레이션을 수행하여 타이머 할당과 유용도 산출에 대한 성능을 해석하였다.

Performance Analysis of Timer Assignment and Utilization of the IEEE 802.4 Token Bus for Real Time Processing

Jeong Ho Kim†, Min Nam Lee†† and Sang Burm Rhee†††

ABSTRACT

The IEEE 802.4 token bus has been widely accepted as the standard for factory local area networks. The priority option of the 802.4 standard supports multiple classes of traffic by using a set of timers to control information exchange. The performance of the 802.4 priority mechanism in industrial real time control is examined. A timer assignment technique is presented for such applications. The timers are set to satisfy the worst case access delay requirements of real time control applications. Other applications that are not time constrainted can be supported simultaneously. In fact, under certain conditions, such applications can also be guaranteed a minimum bandwidth allocation. Simulation results are used to evaluate the timer assignment and utililization.

1. 서 론

대규모 생산 공정에서 제어기기, 로보트들 같은 제조 장치들의 정보 교환은 네트워크에 의존한다. 이 프로그래머블한 장치들간의 메세지 전달은 생산 공정의 실시간 처리를 통하여 원할한 동작을 위해 중요하다.

IEEE 802.4 토큰버스 네트워크는 안정성, 신뢰성, 우선순위의 가능성 때문에 산업 환경에서 네트워크의 표준으로 제안되어 운영되고 있다[1].

생산자동화 프로토콜인 MAP(manufacturing automation protocol)에서도 802.4의 표준을 채택하고 있다[2, 3]. 또한 IEC에 의해 자동화용으로 개발된 PROWAY(process data highway)도 또한 802.4 표준을 채택하고 있다[4].

생산공정에서 실시간 제어 응용을 위해 사용되어지는 네트워크는 다양한 특징과 요구사항을 가지는 다양한 데이타 형태를 전송하게 된다[5]. 공정 제어 메세지는 동기적 또는 비동기적으로 발생될 수 있으나, 이러한 메세지에 허용된 시간 지연은 공정 특성에 따라 다르지만 실시간 제어에 관련된 메세지 전송제한은 20ms에서 50ms 이

† 정 회 원 : 한국전자통신연구소 실장
†† 정 회 원 : 건설공제조합 전산부장
††† 정 회 원 : 단국대학교, 컴퓨터공학과 교수
논문접수 : 1994년 6월 20일, 심사완료 : 1994년 8월 25일

내이다.

또한, 네트워크 매체의 억세스 기능의 우선처리 메카니즘은 네트워크에 연결된 제어시스템과 관련된 실시간 처리 트래픽의 요구를 만족시켜야 한다. IEEE 802.4 표준은 일종의 타이머에 의해 다양한 클래스들간에 대역폭을 할당하는 데 선택적 우선 순위의 메카니즘을 제공한다. 싱글 메세지 클래스를 전송하는 토큰 버스 네트워크의 성능의 특징들은 일반적으로 잘 알려져 있으며, 다중 클래스 트래픽을 수행하는 동일한 802.4 네트워크의 성능은 최근의 여러편의 논문이나 연구과제에서 등장하고 있다[6, 9, 10, 11, 12, 13].

A.P. Jayasuman등은 다중 트래픽 클래스를 수행하는 대칭형 토큰 버스에 대한 처리율 범위를 제시하였고, Watson등은 대칭형 버스 네트워크에서 대기시간, 처리율의 벤치마크 부하패턴을 설정하여 성능의 요구사항을 충족하기 위해 target_rotation_time의 설정의 문제를 해석하였다[10, 11, 13].

본 논문에서는 보다 일반적이면서 부분적인 대칭형 네트워크를 대상으로 하여 주어진 클래스에서 트래픽에서의 타이머 할당구조와 유용도에 대한 해석을 수행한다. 부분적인 대칭형 네트워크에서는 주어진 클래스의 트래픽은 이를 생성하는 장치들간에 대칭적으로 분산되어 있지만 모든 네트워크 노드에서 나타낼 필요는 없다. 또한 시간상 제약이 없는 트래픽 뿐 만 아니라 실시간 처리가 요구되는 트래픽을 전송할 경우 802.4 토큰 버스 네트워크에서 타이머 할당 기법에 의하여 다양한 트래픽 요구를 만족시키는 범위를 해석한다. 그리고 IEEE 802.4 표준의 우선 순위 옵션을 적용하여 다양한 트래픽의 요구사항을 충족하기 위해 타이머 할당 구조를 설정하는 것을 해석하고 이의 시뮬레이션을 통하여 구해진 유용도 결과로서 시간 설정 전략의 한계에 대하여 제시한다.

2. IEEE 802.4 토큰 버스의 우선순위처리

IEEE 802.4 표준은 선택이 가능한 우선 순위 메카니즘을 제공하며 이는 프레임의 우선 순위 처리가 LLC 부계층을 요구함으로서 수행된다. MAC 부계층은 우선 순위 클래스의 4개의 레벨을 제공하며 access_class는 0, 2, 4, 6이라고 불려지고 우선순위처리는 0이 낮고 6이 높으며 우선순위 값을 갖는 프레임을 전송하게 된다. 각 class_access는 토큰이 다음 노드에게 전달되기 전에 모든 access_class를 통하여 가장 높은 access_class로 내부적으로 전달하게 된다. 이와같은 우선 순위 메카니즘의 동작은 다음과 같다.

토큰을 가지고 있는 어느 노드에서 노드 관리에 위해 설정된 hi_pri_token_hold_time(Ts)라 불리어지는 최대 시간을 초과하지 않은 시간안에서 최대의 우선 클래스의 프레임 전송을 초기화하는 것이 허용된다. 이런 hi_pri_token_hold_time(Ts)는 어떤 한 노드가 네트워크를 독점하는 것을 막는다. 우선순위의 가장 높은 프레임을 보낸 후, access_class의 큐를 서비스하기 시작한다. 한 노드에서 세개의 하위 access_class의 각 target_rotation_time 전에 큐에 돌아온다면, 노드는 target_rotation_time이 해제될 때까지 특별한 access_class의 프레임을 전송하는 것이 허용되어진다. 각 클래스의 메세지에 의해 걸리는 시간 지연뿐만 아니라 각 클래스에 할당된 대역폭의 분할은 hi_pri_token_hold_time과 각 access_class의 target_rotation_time에 달려 있다. 하지만 모든 가능한 부하 조건 이하에 있는 대칭적 네트워크에서 타이머 할당은 최상의 우선 순위 클래스와 최대의 target_rotation_time을 가진 access_class에 대해서만 최소한의 대역폭을 보장받게 된다[11]. 이러한 우선 순위처리는 실시간 제어 응용에서는 다양한 트래픽 클래스의 요구사항을 충족하기 위한 hi_pri_token_hold_time과 target_rotation_time의 값을 선정하여야 한다.

3. Target Rotation Timer의 설정

Target rotation timer의 설정을 위하여 다음의

메세지 클래스를 가진 네트워크의 성능을 해석하기로 한다. 클래스 A는 기계의 제어와 다른 실시간 프로세스를 위해 요구되어지는 메세지로 20−50ms 시간 제한안에서 전송되어야 한다. 클래스 B는 제한된 시간의 요구를 갖지 않는 트래픽으로 정적인 네트워크 처리율을 요구하며 데이타 수신과 실시간 프로세스를 제어하지 않은 시스템에 의해 생성된 트래픽이 해당된다. 이의 처리를 위하여 타이머의 설정된 값은 클래스의 메세지에 대해 최소의 보장된 대역폭을 제공한다. 또한 이 두개의 클래스 뿐만 아니라, 네트워크에서 허용된 대역폭을 기초로 하여 파일전송, 데이타베이스 갱신과 관련된 메세지를 전송하는 것이 필요하며 이 트래픽을 클래스 C 라고 한다. 먼저 클래스 A와 클래스 B가 제공된 네트워크를 고려해 본다. 클래스 A 메세지의 요구사항은 MAC 부계층에 의해 제공된 우선순위 6의 서비스를 사용하여 충족된다. 클래스 B의 메세지는 access_class를 사용하여야 한다. 이의 해석에서 네트워크의 운영은 정상으로 가정되고, 논리적 링의 유지 기능과 관련된 오버헤드는 무시한다. 네트워크에서 노드의 수는 논리적 링을 받거나 주는 노드없이 항상 일정하다. 네트워크의 유용도(utilization: U)는 데이타 프레임의 전송 되는 시간의 분배를 말한다. 네트워크의 처리율(S)은 대역폭의 분배에서 시간당 목적지에서 수신되어지는 정보의 총 비트 수로서 정의된다. 클래스 i(i=A,B, 혹은 C)의 유용도와 처리율은 각각 U_i, S_i로 나타내며 이의 관계는 다음과 같다.

$$S_i=\alpha_i U_i \quad \cdots\cdots (1)$$

여기서 $\alpha_i=\overline{T_i}/(T_i+T_{oh})$이며, T_{oh}는 프레임의 오버헤드 부분에 요구되는 시간이다.

클래스 i의 메세지의 평균길이는 T_i라 가정하면 비트 시간으로 표현되는 대응된 전송시간도 T_i가 된다. 싱글비트 전송을 위해 요구되는 시간이 기본적인 단위시간이며 모든 다른 간격들은 비트 시간들로 표현할수 있다. 주어진 부하 G는 채널 대역폭의 분배로서 표현되는 시간당 모든 동작중인 노드에 의해 생성되는 비트의 수이다. 따라서 클래스 i의 주어진 부하(load)는 G_i로 나타낸다. T_i는 두 노드간의 토큰 전달의 평균시간이라 한다. 이것은 노드의 시간지연, 헤드엔드 시간지연, 송신기와 수신기의 모뎀시간지연, 토큰전송시간, 전달지연 시간을 포함한다. 여기에서는 실시간 응용을 위하여 특정한 시간 범위안에서 메세지(클래스−A)를 보내는 것이 필요하다. 이러한 요구사항은 클래스 A의 제한된 시간, 즉 $\widetilde{D_A}$ 안에 이 노드에 토큰이 돌아오는 것을 확인되면 된다. 일단 토큰이 수신되면, 그 노드는 메세지를 전송하기 위해 토큰의 보유시간인 $\widetilde{D_M}$의 시간동안 토큰을 보유하게 되며 이 값은 마지막 토큰 회전 시간 동안에 생성된 모든 메세지를 전송할 수 있는 것이어야 한다. 따라서 제어기는 다른 $\widetilde{D_A}$ 시간 단위 동안에 토큰을 받을 수 없다. 따라서 클래스 A큐의 최대한 억세스시간 지연은 $\widetilde{D_A}$미만으로 보장되어야 한다.

하지만 클래스 A의 메세지 지연 혹은 버퍼 시스템에서 손실의 가능성은 $\widetilde{D_M}$뿐 만 아니라 트래픽 강도에 달려있으므로 최대 메세지의 시간지연은 $\widetilde{D_M}$의 적당한 선택으로 규정된다. 클래스 A의 큐에 대한 최대 억세스, 시간 지연은 연속적인 전송기간 동안에 큐의 시간 간격으로 정의된다. 네트워크에 연결된 클래스 A 장치의 수, $\widetilde{D_A}$와 $\widetilde{D_M}$에 정해진 요구사항에 따라, 클래스 B 의 다른 응용에 최소한의 대역폭을 보장받는 것이 가능하다. 이는 클래스 B의 유용도($\widetilde{U_B}$)에 의해 클래스 B사용과 클래스 B 의 네트워크 ($\widetilde{S_B}$)에 의한 대응된 처리율을 보장받음을 나타낸다. 클래스 B의 응용에서 보장된 처리율을 제공하는 것은 클래스 B에 주어진 부하가 보장된 처리율의 미만이면 모든 클래스 B의 메세지는 한정된 시간지연으로 전달된다. 또한, $\widetilde{S_B}$, $\widetilde{D_A}$와 $\widetilde{D_M}$의 값에 따라, 두개의 메세지 클래스의 요구를 만족시키는 것이 가능하지 않을 수도 있다.

또한, N개의 노드를 가진 토큰 버스 네트워크를 가정하여 한 노드가 클래스 A 혹은 클래스 B 혹은 둘다의 트래픽을 생성할 수 있다고 한다.

N_a와 N_b은 클래스 A 그리고 클래스 B 각각의 메세지를 생성하는 노드의 수이다. 부분적으로 대칭적 네트워크라 하고, 이런 네트워크에서 클래스 A와 클래스 B는 N_a와 N_b의 각 큐들 사이에 동등하게 분산되어 있다고 가정한다. 클래스 A 트래픽을 생성하는 스테이션의 hi_pri_token_hold 시간은 T_s이고, 클래스 B의 트래픽을 생성하는 큐에 대한 target_rotation_time은 T_R이다. 이의 해석은 다양한 클래스의 처리율을 예측하기 위한 토큰 회전 시간의 평균 값을 사용한다.

IEEE 802.4 우선 순위 메카니즘은 타이머 세트를 사용해서 다양한 클래스의 큐에 의해 데이타양을 제어한다. 타이머가 해제될 때에도 다음의 access_class 혹은 토큰을 전송하기 전에 현재의 메세지의 전송을 종료한다. 클래스 A에 대한 오버슈트의 최대 그리고 평균값은 $\widetilde{T_A}$와 T_A로, 클래스 B에 대한 대응된 값들은 각각 $\widetilde{T_B}$와 T_B로 나타낸다. 최대 가능한 시간동안에 클래스 B 노드들이 토큰을 보유하는 토큰 회전 시간의 종료에서 최대 억세스 시간 지연이 있게 되고, 최대 가능한 시간 동안에 클래스 A의 노드들은 토큰을 보유하게 된다. 따라서 주어진 토큰 회전에서 클래스 B의 최대 가능한 토큰 보유시간은 $(T_R - NT_t + \widetilde{T_B})$가 된다. 여기서 N은 논리적 링에서 총 노드의 수이다. 그리고 T_t은 평균 토큰 전달 시간이다.

클래스 A큐 $\widetilde{D_A}$에 대한 최악의 경우 억세스 시간 지연은 다음과 같다.

$$\widetilde{D_A}' = (T_R + \widetilde{T_B}) + (N_A - 1)(T_S + \widetilde{T_A}) \quad \cdots\cdots (2)$$

이런 경우에, 하나의 클래스 B 큐는 $(T_R - NT_t + \widetilde{T_B})$시간 단위동안 토큰을 보유한다. 그리고 최악의 상태시간 지연에 대해서도 클래스 A 큐는 같은 노드에 있다. 다른 $(N_A - 1)$클래스 A 큐의 $(T_S + \widetilde{T_A})$동안 토큰을 보유한다. 토큰 전송의 시간은 NT_t이다. 하지만, 클래스 A 큐에 대한 시간 제약은 각 클래스 A 큐가 $\widetilde{D_A}$단위시간 안에 토큰을 받은 것을 요구한다. 그래서, 타이머 값 T_R과 T_S은 $\widetilde{D_A}' < \widetilde{D_A}$되도록 선택되어야 한다. 즉,

$$(T_R + \widetilde{T_B})(N_A - 1)(T_S + \widetilde{T_A}) \le \widetilde{D_A} \quad \cdots\cdots (3)$$

클래스 A의 큐가 $\widetilde{D_M}$단위시간 동안 토큰을 계속해서 보유하도록 하는 것이 필요하므로 다음의 조건이 만족되어야 한다.

$$T_S \ge \widetilde{D_M} \quad \cdots\cdots (4)$$

토큰 회전 시간 C는 어떤 한 노드에서 access_class가 토큰을 받은 다음 순간까지 경과된 시간으로 정의된다. 여기서 $\overline{C}$를 평균 토큰 회전 시간이라 한다.

U는 네트워크 유용도이고 $(1-U)$는 토큰에 의해 갖는 대역폭의 역이므로 토큰 전송 시간인 NT_t동안 토큰을 전달하는 시간은 $(1-U)\overline{C}$가 된다[8, 9]. 그러므로 네트워크 유용도는 다음과 같이 주어진다.

$$U = 1 - \frac{NT_t}{\overline{C}} \quad \cdots\cdots (5)$$

토큰 회전은 클래스 A의 메세지, 클래스 B의 메세지 그리고 토큰들의 전송을 포함한다. 그래서 평균 토큰 회전 시간 $\overline{C}$는 다음과 같이 주어진다.

$$\overline{C} = N_A U_A \overline{C} + N_B U_B \overline{C} + NT_t \quad \cdots\cdots (6)$$

여기서 U_A와 U_B는 클래스의 A의 큐 그리고 클래스 B의 큐 각각에 의한 네트워크의 평균 유용도이다. 각 사이클에서 클래스 A의 큐가 과도하게 부하가 걸리면, 클래스 A의 큐는 가능한 최대 지연동안 데이타를 전송할 것이다. 즉 $(T_S + T_A)$ 시간 동안이 된다. 따라서 U_A는

$$U_A \le \frac{(T_S + T_A)}{\overline{C}} \quad \cdots\cdots (7)$$

이 된다. 각 클래스 B의 큐는 관계된 target_token_rotation_timer의 해제까지 메세지를 전송하는 것이 허용된다. 타이머가 해제된 후, 현재의 메세지 전송은 토큰이 다음의 access_class에 보내지기 전에 끝마치게 된다. 그러므로,

$$\overline{C} + U_B \overline{C} \le T_R + T_B \quad \cdots\cdots (8)$$

이 된다. 여기서 T_R은 target_rotation 시간이고 T_B는 메세지 전송의 종료 동안 요구되어지는 평균

시간 오버슈트이다.

식 (6)(7)(8)로부터 평균 토큰 로테이션 시간 $\overline{C}$에 대해 상한값을 유도할 수 있다.

즉 $\overline{C} \le C_{max}$이다. 여기서,

$$C_{max} = \frac{1}{(1+N_B)} [N_A(T_S+T_A)+N_B(T_R+T_B)+NT_l] \cdots (9)$$

이며 식(5)를 이용하여, 네트워크 유용도에 대해 대응된 상한값은 다음과 같이 주어진다.

$$U_{max} = 1 - \frac{NT_l(1+N_B)}{[N_A(T_S+T_A)+N_B(T_R+T_B)+NT_l]} \cdots\cdots (10)$$

클래스 B 큐에 보장된 유용도 요소는 식(7)과 식(10)을 이용하여 얻어지고, 식 (11)과 같이된다.

$$U_B = U_{max} - \frac{N_A(T_S+T_A)}{C_{max}} \cdots\cdots (11)$$

그러므로, 클래스 B의 요구사항을 만족하기 위해 타이머는 $U_B > \widetilde{U_B}$되기 위해 선택되어져야 한다. 식 (9)와 식(11)을 사용하여

$$(T_R+T_B) > \left\{1+\frac{\widetilde{U_B}}{N_B}\right\} \frac{[NT_l+N_A(T_S+T_A)]}{(1-\widetilde{U_B})} \cdots\cdots (12)$$

이 되므로 클래스 B에 대한 대역폭을 보장하기 위해 식(12)를 만족하여야 한다. 클래스 A의 억세스시간 지연 요구사항은 식(3), (4)를 만족하면 충족된다. 따라서 네트워크는 식(3)(4)와 식(12)가 동시에 만족되기만 하면 클래스 A의 트래픽4을 생성하는 N_A와 클래스 B 트래픽을 생성하는 N_B노드를 지원할 수 있다. 식 (12)에 의해 제시된 조건이 참이면, 클래스 B는 $\widetilde{U_B}$의 정상 네트워크 유용도를 수신할수있다. 따라서 클래스 B 큐는 (N_B+1)토큰 회전마다 최소한 하나의 메세지를 전달하는 기회를 갖는다.

위의 분석은 클래스 A와 B의 메세지만이 제공되는 네트워크만으로 해석하였다. 그러면 다음의 단계는 클래스 A와 B의 요구를 인정하고 클래스 C 트래픽을 네트워크에 적용한다면 클래스 C 메세지는 유용한 대역폭에 기초하여 엄격하게 전송되어지고, 802.4 MAC 부계층의 억세스 클래스 2 혹은 억세스 클래스 0을 경유하여 다루어진다. 클래스 C에 대한 target_rotation_time을 T_R'라 하자. T_R'가 $T_R' << T_R$이기 위해 클래스 C의 메세지는 토큰 회전 시간이 충분히 T_R보다 작을 때만 전송되어질 것이다. 클래스 C의 메세지 서비스는 토큰 회전 시간 타이머 오버 슈트 때문에 T_R을 초과하지 않도록 한다.

4. 시뮬레이션 결과 및 검토

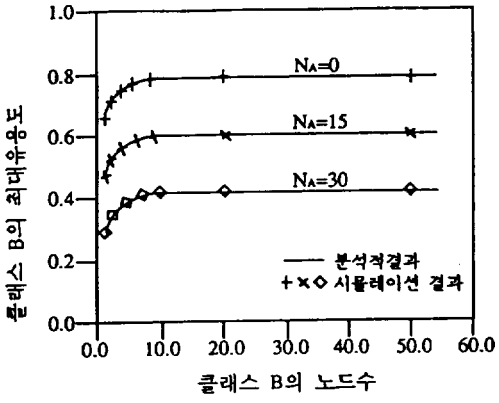
IEEE 802.4 우선 순위 처리에 대한 시간할당구조와 이의 성능해석을 위하여 사용된 네트워크 파라미터는 다음과 같다.

노드의 수 N:	50
평균 데이타 필드의 길이 T_M:	1204 bits
프레임당 오버헤드 T_{oh}:	184 bits
토큰의 길이:	184 bits
버스의 대역폭:	10 Mbit/s
한 방향으로의 전달 지연시간:	5.0μs
헤드 엔드와 모뎀지연시간:	10.5μs
노드 시간 지연:	50.0μs
평균 토큰 패싱 시간 T_l:	83.5μ
Hi_pri_token_hold_time T_S:	0.2ms
클래스 4의 토큰 회전 시간:	20.0ms

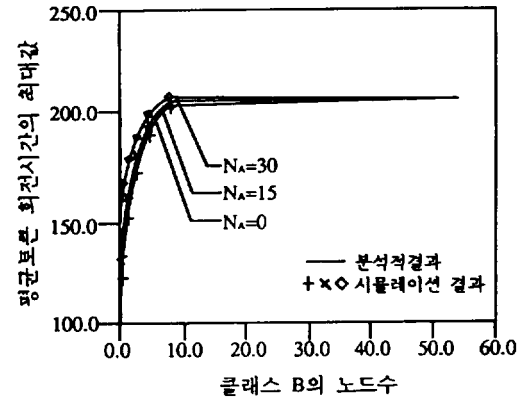
클래스 A의 메세지는 일정한 도착 간격 시간과 일정한 메세지 길이로 특징되어지며, 클래스 B와 C의 메세지는 Poisson 분포에 따라 도착하며 클래스 B의 데이타 필드의 길이는 평균 T_M으로 지수적으로 분포되어지고, $4T_M$보다 긴 메세지는 $4T_M$으로 전달된다. 각 메세지는 이렇게 변하는 데이타필드 길이 뿐만 아니라 T_{oh}의 고정길이의 헤더를 갖는다. 클래스 메세지의 길이는 지수적으로 분포되는 것으로 한다.

먼저, IEEE 802.4의 타이머 할당구조 해석을 위하여 802.4 우선순위구조에서 클래스 A 의 노드수 N_A와 클래스 B 의 노드수 N_B의 값을 변화시켜 $Cmax$와 $\widetilde{U_B}$를 구하도록 시뮬레이션 하였다.

그림 1은 다양한 N_A값(N_A=0, 15, 30)에 대해서 N_B에 대한 유용도 U_B의 변화를 나타내었다. 클래스 B의 처리율에 대한 최대값과 평균 토큰 회전 시간은 과대 부하 조건에서 네트워크를 시뮬레이션 함으로써 구할수 있다. 이러한 조건하에서, 각 큐는 메세지 전송이 token_hold_timer가 해제될 때까지 계속하기 위해 그 안에 충분한 메세지를 가진다. 식(9)와 식(11)을 사용하여 얻어진 U_B와 $Cmax$에 대한 분석한 값을 (그림 1)과 (그림 2)



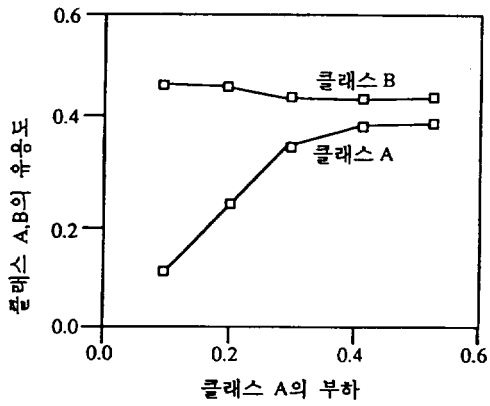
(그림 1) 클래스 B의 노드수(N_B)와 유용도(U_B)관계
(Fig. 1) Variation of the utilization for class B versus number of class B nodes



(그림 2) 클래스 B의 노드수(N_B)와 평균토큰 회전시간 (C_{max} X 10_{ms})의 최대값 관계
(Fig. 2) Variation of the maximum value of the mean token_rotation_time(Cmax X 10ms) with the number of class B nodes

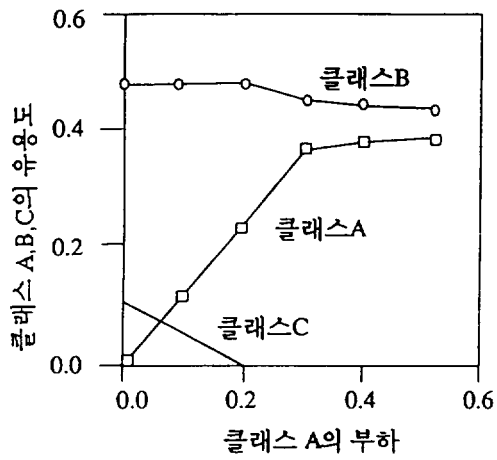
에 나타내었다. 최대 토큰 회전 시간 뿐만 아니라 클래스 B에 보장된 대역폭은 N_A와 N_B에 대한 전 영역의 값에 대해 분석 모델에 의해 예측된 값과 매우 잘 일치함을 알 수 있다.

다음은 네트워크 파라미터 N_A, N_B, T_R과 T_S에 대한 네트워크 유용도를 해석한다. 클래스 B의 유용성은 또한 클래스 B의 노드의 수에 의존한다. 하지만 이런 의존성은 N_B가 클 때는 중요하지 않으며, 클래스 B에 유용한 대역폭은 클래스 B 트래픽을 발생시키는 N_B노드에 의해 공유된다. 다른 파라미터들을 일정하게 하면서 클래스 B의 target_rotation_time의 증가는 네트워크의 총 유용도를 증가시키게 되는데, 이는 평균 토큰 회전 시간의 최대값이 증가되는 것을 의미하며, 이로 인해 클래스 A 메세지의 최악의 경우로서 억세스 시간지연을 증가시키게 된다. 클래스 A의 큐 N_A(혹은 hi_pri_token_hold_time T_S)의 수를 증가시키는 것은 클래스 A 메세지에 유용한 더 넓은 대역폭을 갖게되며 클래스 B에 보장된 유용성을 감소시킨다. 클래스 N_A=N_B=30을 가진 다중 클래스의 트래픽을 수행하는 토큰버스 네트워크의 성능 평가에서 클래스 B에 주어진 부하는 0.4의 상수값을 갖고, 클래스 A의 주어진 부하는 증가된다. 주어진 부하는 채널 대역폭으로 정규화하고, 어드레스 필드와 같은 오버헤드를 포함하지 않았다. 클래스 A의 주어진 부하에 대한 클래스 A,B에 대한 유용성을 (그림 3)에 나타내었다. 여기에서 보면, 클래스 B의 유용성이 감소되면서 클래스 A의 트래픽을 조정하고 있다. 하지만 클래스 A에 있는 트래픽의 증가는 클래스 B의 유용성이 0.419이하로 감소하지 않도록 하고 있으며 이는 식(11)에 의해 예측된 0.424의 보장된 대역폭과 잘 일치함을 알수있다. 다음에 클래스 C의 경우 트래픽 강도는 0.1의 상수 값을 갖고, 클래스 B는 0.4를 갖는다. 클래스 C는 MAC 부계층에서 우선순위 2로 할당되고, 클래스 2의 target_rotation_time은 10ms이다. 클래스 A에 트래픽이 증가할 때 주어진 부하 특성에 대한 유용성은 (그림 4)에 나타내었다. 이 경우에



(그림 3) 클래스 B의 부하(GB)가 일정할 경우 클래스 A의 주어진 부하(GA)에 대한 클래스 A와 B의 유용도(UA, UB)

(Fig. 3) The utilization of class A and class B versus the offered load of class A with the offered load in class B held constant

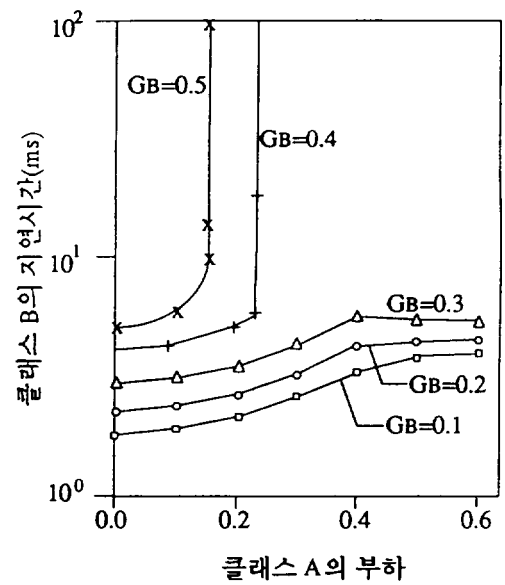


(그림 4) 클래스 B의 부하(G_B), 클래스 C의 부하(G_C)가 일정할 경우 클래스 A, B와 C의 유용도 (U_A, U_B, U_C)

(Fig. 4) The utilization of class A, class B, and class C versus the offered load of class A with the offered load in class B and class C held constant

대해 클래스 A와 B의 유용성 특성은 (그림 3)의 것과 유사하다. 그러므로 클래스 C에 대한 target_rotation_time이 B의 것보다 작도록 잘 선택되고, 클래스 시간 T_K이상의 토큰 회전 시간을 증가시킬 정도로 길지 않으면, 메세지는 유효한 대역폭내에서 엄격하게 다루어질수 있다. 클래스 A와 B에 선택된 타이머 값은 클래스 B에 대해 0.424, 클래스 A에 대해 0.365의 유용성의 값을 갖는데 이것은 각 0.359, 0.309의 처리율 값과 대응된다. 클래스 A 혹은 B의 주어진 부하가 그 클래스에 대한 보장된 처리율 미만이면, 그 클래스에서의 메세지의 시간 지연은 다른 클래스에서의 주어진 부하와 함께 증가되어서는 안된다. (그림 5)와 (그림 6)의 시뮬레이션 결과는 이것이 타당함을 보여준다.

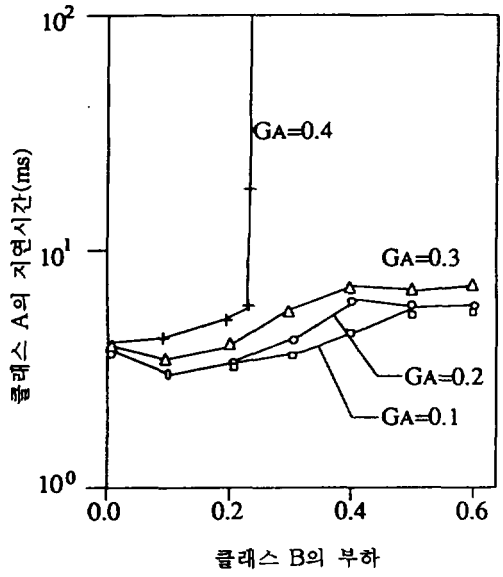
(그림 5)는 클래스 B에서 주어진 부하가 일정할 때, 클래스 A의 주어진 부하와 클래스 B의 평균시간 지연을 나타내고 있다. G_B=0.1, 0.2, 0.3에 대해, 클래스 A의 부하의 증가에도 불구하고, 클래스 B 메세지의 시간지연은 경계지을 수 있다. 즉, 클래스 B에서 주어진 부하가 보장된 처리율보다 더 크면, 클래스 B의 메세지의 평균 시간 지연은 클래스에서의 부하의 증가와 함께 무한정 증가된다. 이런 경우에, 도착된 메세지의



(그림 5) 클래스 B의 부하(G_B)가 일정할 경우 클래스 A의 부하(G_A)에 대한 클래스 B메세지의 평균지연시간

(Fig. 5) Mean delay of class-B messages(ms) versus the offered load of class A with offered load of class B constant

전부가 전달되는 것이 아니고, 클래스 B의 큐의 길이는 제한 없이 증가될 것이다. 하지만, 클래스 B는 대역폭 할당구조의 결과로서 일정한 네트워크 처리율을 여전히 받을 것이다. 한정된 버퍼 시스템에서, 그러한 조건은 전송되어질 메세지의 한정된 시간지연의 결과를 가져오고, 버퍼 오버플로우 때문에 메세지의 손실의 가능성을 가져온다.



(그림 6) 클래스 A의 부하(G_A)가 일정할 경우 클래스 B의 부하(G_A)에 대한 클래스 G_A메세지의 평균지연시간

(Fig. 6) Mean delay of class A messages(ms) versus the offered load of class B with offered load of class A constant

(그림 6)은 클래스 B의 주어진 부하와 함께 클래스 A 메세지의 평균시간 지연의 변화를 나타낸다.클래스 A에 대한 보장된 처리율인 0.309 보다 클래스 B의 부하의 증가는 무절제한 클래스 A의 메세지의 시간지연을 가져오지는 않는다. 이것은 G_A=0.4인 경우에 대해 클래스 B의 부하의 증가와 함께 클래스 A 메세지의 평균시간 지연의 증가를 나타낸다.

5. 결 론

IEEE 802.4의 우선 순위 처리 메카니즘은 토큰 버스 네트워크에 대한 다중 클래스 네트워크를 지원할수 있다. 우선순위구조와 관계된 파라미터, 즉 hi_pri_token_hold_time과 target_token_rotation_time이 다중 클래스에서 제공되는 서비스의 질을 결정하며 이에따른 타이머 할당은 트래픽의 다양한 요구를 충족시키기 위한 타이머 값을 결정하는데 사용될 수 있다.

본 논문에서는 보다 일반적이면서 부분적인 대칭형 네트워크를 대상으로 하여 주어진 클래스에서 트래픽에서의 타이머 할당구조와 유용도에 대한 해석을 하였다. 이는 하나의 메세지의 클래스를 생성하는 한 노드와 세개의 메세지 클래스를 생성하는 다른 한 노드의 상황에서 클래스마다의 정상 네트워크에서 타이머 할당에 대한 상한값을 유도하는 식을 설정하여 유용한 대역폭을 해석하였다. 또한 유용도에 대한 세종류의 메세지 클래스에 대해서는 시뮬레이션에 의해 얻어진 결과는 제안된 타이머 할당 구조에 따른 해석식을 사용하여 얻어진 분석값과 매우 잘 일치하였다. 이는 어떤 실시간 제어 프로세스들은 작은 억세스 시간 지연을 요구하는 짧은 메세지를 생성하는 것이 가능하지만 어떤 다른 실시간 프로세스는 긴 메세지를 생성하는 경우 상대적으로 보다 긴 억세스 시간 지연을 허용할 수 있음을 알수있다.

제안된 타이머 할당구조는 각 클래스는 어떤 제한된 시간안에 토큰을 수신하며 이는 어떤 제한 시간동안 토큰을 유지할 수 있도록 보장한다. 그러나 클래스 A의 각 패킷의 억세스 시간지연은 노드의 트래픽 강도(intensity)면에서 검토할 필요가 있다.

또한 네트워크에서는 잃어버린 토큰 그리고 다중 토큰으로부터 회복 그리고 다른 논리적 링의 유지 기능들이 수행되어져야 하므로 이러한 오버헤드가 고려된 조건 아래에서 802.4 우선순위 메카니즘의 성능평가 연구가 필요하다.

참 고 문 헌

[1] ANSI/IEEE std. 802. 4-1985:Token Passing

Bus Access Method and Physical Layer Specifications, IEEE, 1985.

[2] Manufacturing Automation Protocol, Version 3.0, GM Tech. Center, Mar. 1988.

[3] L. J. McGriffin, L. O. Reid, and S. R. Sparks, "MAP/TOP in CIM Distributed Computing", IEEE Network, Vol. 2, No. 3, pp. 23-31, May 1988.

[4] IEC Process Data Highway(PROWAY C), International Electrotechnical Commission (IEC) Publication 955, Jan. 1985.

[5] A. Ray and Y. Halevi, "Finite-Dimensional Modeling of Network induced Delays for Real-Time Control Systems", in Proc. Amer. Contr. Conf., pp. 1077-1082, Jun. 1988.

[6] S. R. Sachs, K. Kan, and J. A. Silvester, "Performance Analysis of a Token-Bus Protocol and Comparison with Other LAN Protocols", in Proc. 10th Conf. Local Comput. Networks, pp. 46-51, 1985.

[7] M. E. Ulug, "Comparison of Token-Holding Strategies for a Static Token-Passing Bus", in Proc. Comput. Networking Symp., pp. 37-44, 1984.

[8] M. A. Colvin and A. C. Weaver, "Perfomance of Single Access Classes on the IEEE 802.4 Token Bus", IEEE Trans, Commun., Vol. COM-34, No. 12, pp. 1253-1256, Dec. 1986.

[9] A. P. Jayasumana, "Comment on Performance of Single Access Classes on the IEEE 802.4 Tokes Bus", IEEE Trans. Commun., Vol. COM-36, No. 2, pp. 224-225, Feb. 1988.

[10] A. P. Jayasumana, and P. D. Fisher, "Performance Modeling of IEEE 802. 4 Token Bus", in Proc. IEEE-NBS Workshop Factory Commun., pp. 221-252, Mar. 1987.

[11] A. P. Jayasumana, "Throughput Analysis of IEEE 802.4 Priority Scheme", IEEE Trans. Commun., Vol. 37, No. 6, pp. 565-571, Jun. 1989.

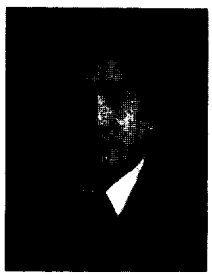
[12] M. Gorur and A. C. Weaver, "Setting Target_Rotation_Times in an IEEE Token Bus Network", IEEE Trans. Ind. Electron., Vol. 35, No. 3, pp. 366-371, Aug. 1988.

[13] D. Janetzky and K. S. Watson, "Performance Evaluation of the MAP Token Bus", in Advances in Local Area Networks, Limb, Kummerle, and Tobagi, eds. New York:IEEE, pp. 411-425, 1987.

[14] FDDI Token Ring Media Access Control, Draft Proposed American National Standard, Fed. 28, 1986.

[15] A. P. Jayasumana and G. G. Jayasumana, "Simulation and Performance Evaluation of 802.4 Priority Scheme", in Proc. IEEE Symp. Simulation Comput. Networks, pp. 232-238, Aug. 1987.

[16] A. P. Jayasumana, "On Use of 802.4 Token Bus in Distributed Control Systems", in Proc. Amer. Contr. Conf., pp. 1094-1099, Jun. 1988.

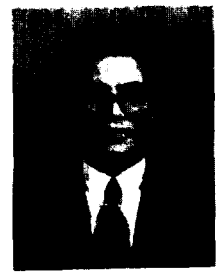


김 정 호

1980년 2월 경북대학교 전자공학과 (학사)
1983년 2월 경북대학교 대학원 전자공학과 (석사)
1990년 8월 전자계산조직응용기술사
1991년 8월 공업계측제어 기술사
1992년 12월 전기통신 기술사
1983년－현재 한국전자통신연구소 책임연구원/실장
관심분야 : 컴퓨터네트워크, 실시간처리시스템

이 민 남

1980년 2월 동국대학교 전자계산학과 (학사)
1982년 2월 연세대학교 대학원 전자계산학과 (석사)
1990년 8월 정보관리 기술사
1991년－현재 건설공제조합 전산부장
관심분야 : 프로젝트관리기법, 컴퓨터네트워크, 데이타베이스설계



이 상 범

1974년 2월 연세대학교 전자공학과 (학사)
1978년 8월 서울대학교 대학원 전자공학과(석사)
1986년 2월 연세대학교 대학원 전자공학과(박사)
1983년 10월－1984년 12월 미국 IOWA 대학 컴퓨터 공학과 객원교수
1979년－1992년 단국대학교 전자공학과 교수
1993년－현재 단국대학교 컴퓨터공학과 교수
관심분야 : 컴퓨터구조, 마이크로프로세서 응용, 영상처리시스템