패킷 스케쥴링을 위한 결손 보완 계층적 라운드로빈 알고리즘

(A Hierarchical Deficit Round-Robin Algorithm for Packet Scheduling)

편 기 현 † 조 성 익 †† 이 종 열 ††
(Kihyun Pyun) (Sung-Ik Cho) (Jong-Yeol Lee)

요 약 지난 십여년동안 각 세션에게 대역폭을 공평하게 분배하기 위한 많은 연구가 수행되었다. 이 문제에 있어서 가장 중요한 도전은 확장성 있는 구현(scalable implementation)을 실현하면서도 동시에 높은 공평성을 성취하는 것이다. 여기서 높은 공평성이란 작은 시간 구간에 대해서도 대역폭이 공평하게 분배되는 것이다. 불행히도 현존하는 스케쥴링 알고리즘들은 확장성 있는 구현에 문제점이 있거나 혹은 공평성이 현저히 낮다는 결점을 갖고 있다. 본 논문에서 우리는 확장성을 잃지 않으면서도 동시에 합리적인 수준의 공평성을 제공하는 패킷 스케쥴링 알고리즘을 제안한다. 제안하는 알고리즘은 결손을 보완하는 계층적 라운드-로빈 알고리즘이다. 계층적 라운드-로빈 알고리즘은 구현 복잡도가 상수 시간인 반면, 성취할 수 있는 공평성은 PGPS(Packet-by-Packet Generalized Processor Sharing) 알고리즘과 비슷함을 보인다. PGPS 알고리즘은 N을 세션 수라고 할 때 정렬된 우선 순위 큐를 사용하기 때문에 O(log N) 구현 복잡도를 가지므로 확장성이 떨어진다.

키워드 : 패킷 스케쥴링, 공평 서비스

Abstract For the last several decades, many researches have been performed to distribute bandwidth fairly between sessions. In this problem, the most important challenge is to realize a scalable implementation and high fairness simultaneously. Here high fairness means that bandwidth is distributed fairly even in short time intervals. Unfortunately, existing scheduling algorithms either are lack of scalable implementation or can achieve low fairness. In this paper, we propose a scheduling algorithm that can achieve feasible fairness without losing scalability. The proposed algorithm is a Hierarchical Deficit Round-Robin(H-DRR). While H-DRR requires a constant time for implementation, the achievable fairness is similar to that of Packet-by-Packet Generalized Processor Sharing(PGPS) algorithm. PGPS has worse scalability since it uses a sorted-priority queue requiring O(log N) implementation complexity where N is the number of sessions.

Key words : packet scheduling, fair service

1. 서 론

지난 십여년동안 각 세션에게 대역폭을 공평하게 분배하기 위한 많은 연구가 수행되었다[1-8]. 이 문제에 있어서 가장 중요한 이슈는 확장성 있는 구현(scalable implementation)을 실현하면서도 동시에 높은 공평성을 성취하는 것이다. 여기서 높은 공평성이란 작은 시간 구간에 대해서도 대역폭이 공평하게 분배되는 것이다.

불행히도 현존하는 스케쥴링 알고리즘들은 확장성 있는 구현에 문제점이 있거나 혹은 너무 넓은 시간 구간에 대해서만 대역폭이 골고루 분배되는 결점을 갖고 있다. 예를 들면, 타임스탬프(time-stamp)에 기반한 알고리듬들은 정렬된 우선 순위 큐를 사용해서 패킷을 전송한다[1-5,9-11]. 정렬된 우선 순위 큐는 시퀀서(sequencer)[12] 혹은 시스톨릭 어레이(systolic array)[13]와 같은 특별 하드웨어를 사용하면 O(1) 복잡도로 구현할 수 있다. 그러나, 이 하드웨어는 라우터 가격을 높인다. 또

· 이 논문은 2004년도 한국학술진흥재단의 지원에 의하여 연구되었음 (KRF-2004-003-D00240).
† 종신회원 : 전북대학교 전자정보공학부 교수
khpyun@chonbuuk.ac.kr
†† 비 회 원 : 전북대학교 전자정보공학부 교수
sicho@chonbuk.ac.kr
jong@chonbuk.ac.kr
논문접수 : 2004년 5월 24일
심사완료 : 2004년 11월 22일

한, 그러한 하드웨어가 수만개의 세션들을 지원할 수 있는지 명확하지 않다. 대조적으로, DRR(Deficit Round Robin)[14-17]과 같은 라운드-로빈 알고리즘들은 확장성 있는 구현을 성취하기 위해서 서비스 순서를 고정한다. 그러나, 현존하는 라운드-로빈 알고리즘들은 공평성 성취에 대한 근본적인 제한점, 즉, 모든 세션들이 한번의 서비스 라운드를 완료하는 시간에 대해서만 대역폭이 골고루 분배되는 특징을 갖는다. 이 때 공평성은 최악의 경우 발생할 수 있는 서비스 라운드 구간의 길이에 반비례하는데, 그 길이의 값이 일반적으로 매우 크기 때문에 결과적으로 매우 낮은 공평성을 성취할 수 있다[4].

본 논문은 확장성을 잃지 않으면서도 동시에 좁은 시간 구간에 대해서도 대역폭을 골고루 분배하는 패킷 스케쥴링 알고리즘을 제안한다. 제안하는 알고리즘은 결손 보완 계층적 라운드-로빈(Hierarchical Deficit Round-Robin) 알고리즘이다. 계층적 라운드-로빈 알고리즘은 구현 복잡도가 상수 시간인 반면, 성취할 수 있는 공평성은 PGPS(Packet-by-Packet Generalized Processor Sharing) 알고리즘과 비슷함을 보인다. PGPS 알고리즘은 N을 세션 수라고 할 때 정렬된 우선 순위 큐를 사용하기 때문에 O(log N) 구현 복잡도를 가지므로 확장성이 떨어진다. 제안하는 알고리즘은 DRR 알고리즘과 비교하면 같은 구현 복잡도를 가지면서도 현저히 높은 공평성을 제공함을 실험을 통해 보인다.

라운드-로빈 알고리즘을 계층적으로 사용하는 기본 아이디어는 과거에도 존재하였다[15-17]. 그러나, 높은 공평성을 성취하기 위해서 계층을 어떻게 구성하고 사용하는 지에 대한 연구와 분석은 본 논문의 중요한 공헌 중 하나이다. 그 중 한 분석 결과로, 우리는 제안하는 라운드-로빈 알고리즘이 패킷 크기가 작고 고정된 ATM 망에서 더욱 높은 공평성을 제공할 수 있음을 증명하였다. 또, 우리는 [14]에서 제시한 DRR이 일반적인 경우에 높은 공평성을 성취할 수 없음을 수학적으로 증명하고 실험을 통해서도 보인다.

본 논문의 구성은 다음과 같다. 2절은 제안하는 결손 보완 계층적 라운드-로빈 알고리즘을 기술하고, 동작 방법을 설명한다. 3절은 결손 보완 계층적 라운드-로빈 알고리즘이 어떻게 높은 공평성을 성취하는 지를 수학적으로 입증한다. 4절은 실험을 통해서 결손 보완 계층적 라운드-로빈 알고리즘이 대역폭을 공평하게 분배함을 보인다. 마지막으로, 5절은 이 논문의 결론을 맺는다.

2. 결손 보완 계층적 라운드-로빈 알고리즘

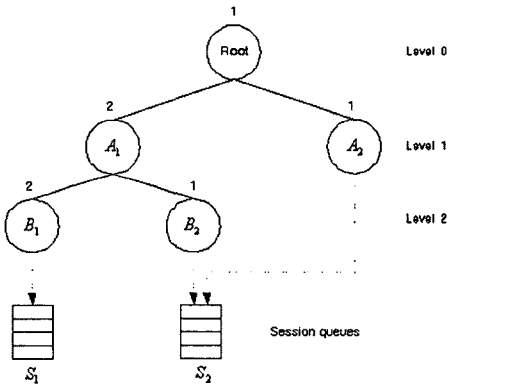
결손 보완 계층적 라운드-로빈 알고리즘은 세션 큐들 위에 링크-공유 계층(link-sharing hierarchy)을 갖는 데이타 구조를 유지한다. 이 계층의 0번째 단계에는 뿌리 노드(root node)라 불리는, 전송 링크를 나타내는 노드 한 개가 존재한다. 각 노드는 가중치를 가지며 부모 노드(parent node)로부터 자신의 가중치에 비례해서 부모 노드의 대역폭을 분배 받는다. 잎이 아닌 노드(non-leaf node)는 부모 노드로부터 자식 노드들(children nodes)에게 대역폭을 분배하는 목적을 갖는다. 반면 잎 노드(leaf node)는 실제로 세션 큐를 서비스하는데 사용된다. 따라서, 오직 잎 노드만이 세션 큐에 대한 포인터를 갖는다. 우리는 여러 개의 잎 노드들이 하나의 세션 큐를 가르키는 것을 허용하는데, 세션이 필요로 하는 대역폭 만큼 꼭 맞게 할당하기 위함이다.

우리는 세션 큐에 적어도 하나 이상의 패킷이 있으면 그 잎 노드는 저장 기간(backlogged period)내에 있다고 말한다. 한 패킷이 비어 있는 세션 큐에 도착할 때, 그 큐를 가르키는 모든 잎 노드들이 새롭게 저장 기간 내에 있게 된다. 또한, 우리는 잎이 아닌 노드의 모든 후손 노드들(descendent nodes) 중에 하나라도 저장된 잎 노드가 존재하면 그 잎이 아닌 노드가 저장 기간내에 있다고 정의한다. 새롭게 저장 기간 내에 있게 된 잎 노드는 그 잎 노드로부터 뿌리 노드의 경로상에 있는 모든 노드들을 새롭게 저장 기간 내에 있도록 만들 수도 있다.

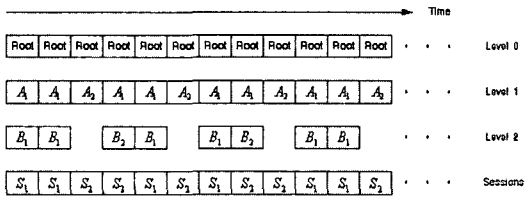
결손 보완 계층적 라운드-로빈 알고리즘은 다음과 같이 정의하는 계층적 라운드-로빈 순서로 저장 기간 내에 있는 노드들을 서비스한다. 만일 뿌리 노드가 저장 기간 내에 있으면 서비스 슬롯(service slot)들이 뿌리 노드에 하나씩 주어진다. 한 서비스 슷롯은 고정된 크기의 패킷양을 전송 링크 속도로 전송하는데 필요한 시간으로 정의된다. 만일 뿌리 노드와 같이 잎이 아닌 노드가 서비스 슬롯을 받으면, 그 슬롯은 가중치를 고려한 라운드-로빈 순서로 저장 기간 내에 있는 자식 노드들에게 전달된다. 형제 노드들(sibling nodes) 사이의 서비스 슬롯 전달 순서는 저장 기간 내에 있게 된 순서에 의해서 결정된다. 결국, 서비스 슬롯은 잎 노드에 도착하게 되고, 그 잎 노드는 그 슬롯을 소모해서 자신이 가르키고 있는 세션 큐에 쌓여 있는 패킷들을 전송한다.

각 노드 n은 양의 정수인 가중치 w_n을 갖는다. 뿌리 노드는 오직 한 개만 존재하기 때문에 양의 정수인 임의의 가중치 값을 사용해도 아무런 차이가 없기 때문에 편의상 항상 1의 값을 갖는 것으로 정한다. 또한, 우리는 형제 노드들의 가중치 합에 제한을 가한다. 단계 l에 위치한 노드 n을 고려할 때, 이 노드의 자식 노드들의 가중치 합은 상수 값 N_l을 초과하지 않도록 제약을 가한다. 따라서, 만일 노드 n이 저장 기간 내에 있다면, 이 노드는 $N_0*\cdots*N_{l-1}$ 개의 서비스 슬롯이 뿌리 노드에 주어질 때 마다 적어도 한 번의 슬롯을 할당받는다.

그림 1은 링크 공유 계층에서 노드들에게 서비스 슬롯이 분배되는 순서를 예를 들어 보여준다. 이 계층에서, 뿌리 노드는 두 자식 노드, A_1과 A_2를 갖는다. 노드 A_1은 두 자식 노드, B_1과 B_2를 갖는다. 노드 B_1은 세션 S_1의 큐를 가르킨다. B_2와 A_2 양쪽 노드는 동시에 세션 S_2의 큐를 가르킨다. A_1과 B_1 노드의 가중치는 2이다. B_2와 A_2 노드는 가중치 1을 갖는다. 세션 S_2가 세션 S_1이 저장 기간 내에 있게 된 이후 바로 저장 기간 내에 있게 되고, 이 두 세션이 계속해서 저장 기간 내에 있다고 가정하자. 그러면, 루트 노드에 주어지는 서비스 슬롯은 다음의 순서대로 세션들에게 전달된 후 소모된다. $S_1,S_1,S_2,S_2,S_1,S_1,S_2,S_2,S_1,S_1,S_2,\cdots$.



(a)



(b)

그림 1 (a) 링크-공유 계층 예. (b) 주어진 서비스 슬롯이 노드 및 세션들에게 주어지는 순서

각 잎이 아닌 노드는 빈 세션 큐를 살피는 것을 피하기 위해서 저장 기간 내에 있게 된 자식 노드들의 색인 목록, 즉, 서비스 목록을 유지한다. 잎이 아닌 노드가 서비스 슬롯을 받을 때 마다 항상 자신의 서비스 목록의 처음에 기재되어 있는 머리 노드(head node)에게 그 슬롯을 전달한다. 또한, 서비스 목록은 머리 노드가 갖는 가중치와 동일한 수의 슬롯이 할당되면 서비스 목록의 마지막에 그 노드의 색인을 이동시킨다. 만일 머리 노드가 서비스 슬롯을 소비했을 때 머리 노드가 가르키는 세션 큐가 비게 되면 서비스 목록에서 그 노드의 색인을 삭제한다. 다음으로, 세션이 새롭게 저장 기간 내에 있게 되었을 때 서비스 목록이 어떻게 갱신되는 지를 살펴 보자. 만일 한 노드가 새롭게 저장 기간 내에 있게 되면, 부모 노드의 서비스 목록에 그 노드의 색인이 첨부된다. 그림 2는 위의 예제에서 서비스 목록이 어떻게 갱신되는 지를 보여준다. 모든 서비스 목록들은 그림 2(a)에 보인바와 같이 초기에는 비어 있다. 세션 S_1이 새롭게 저장 기간 내에 있게 되면, 잎 노드 B_1이 세션 S_1의 큐를 가르키고 있으므로 부모 노드인 노드 A_1의 서비스 목록에 노드 B_1에 대한 색인이 첨부된다. 또한, 새롭게 저장 기간 내에 있게 되는 노드 A_1 때문에 그 부모 노드인 뿌리 노드의 서비스 목록에 노드 A_1에 대한 색인이 추가된다. 그림 2(b)는 세션 S_1이 새롭게 저장될 때 그 갱신된 서비스 목록들을 보여준다. 세션 S_2는 세션 S_1이 저장 기간 내에 있게 된 이후에 바로 저장 기간 내에 있게 된다. 세션 S_2의 큐는 노드 B_2와 A_2 양쪽 노드에 의해서 모두 가르켜지므로, 노드 B_2에 대한 색인과 노드 A_2에 대한 색인이 각각의 부모 노드인 A_1과 뿌리 노드의 서비스 목록에 추가된다. 이 때 노드 A_1의 부모 노드인 뿌리 노드는 이미 노드 A_1에 대한 색인을 갖고 있으므로, 뿌리 노드의 서비스 목록에 A_1에 대한 색인을 추가로 첨부 하지 않는다. 그림 2(c)는 세션 S_2가 새롭게 저장 기간 내에 있게 될 때 갱신된 서비스 목록들을 보여준다.

서비스 목록의 항목들은 세션 큐가 비게 될 때 제거된다. 세션 큐가 비게 될 때 그 세션 큐를 가르키는 잎 노드는 저장 기간 내에 있지 않으므로 부모 노드의 서비스 목록에서 그 잎 노드에 대한 색인을 제거 한다. 만일 잎이 아닌 노드의 서비스 목록에 존재하는 항목들이 모두 삭제되면, 그 부모 노드의 서비스 목록에 존재하는 그 잎이 아닌 노드의 색인을 재귀적으로 삭제한다.

서비스 목록의 머리 노드는 자신의 가중치와 동일한 수의 서비스 슬롯을 할당받으면 맨 뒤쪽으로 이동된다. 각 노드 n은 자신이 받은 슬롯 수를 기록하는 서비스 카운터 c_n을 유지한다. 그림 3은 그림 2(c)에서 보인 서비스 목록들에게 서비스 슬롯이 주어질 때 어떻게 각 서비스 목록들이 갱신되는 지를 보여준다.

결손 보완 계층적 라운드-로빈 알고리즘은 한 서비스 슬롯 동안 평균 고정된 크기 Q 바이트를 전송한다. Q는 최대 패킷 크기 L^{max} 보다 작지 않은 정수로 설정된다. 패킷 크기는 가변이므로 잎 노드는 매번 주어진 슬롯마다 정확히 Q 바이트씩 정확하게 전송할 수 있는

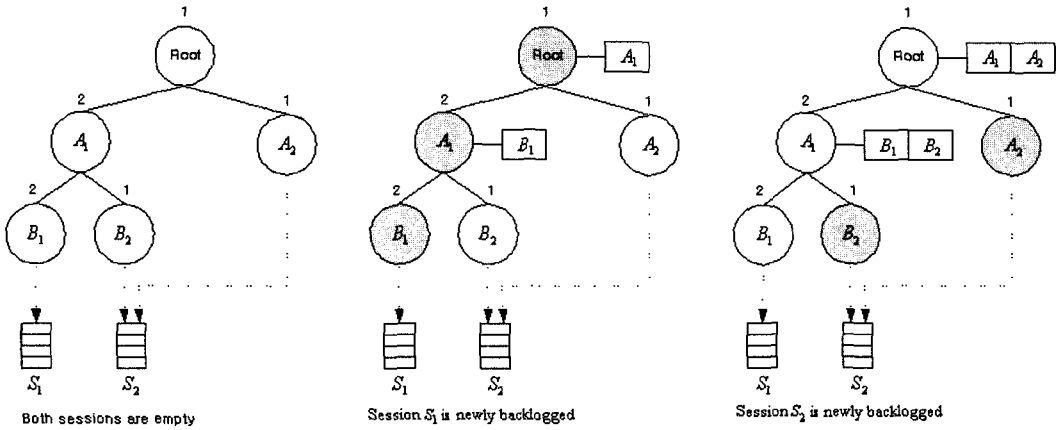


그림 2 (a) 초기에 모든 잎이 아닌 노드들, 즉, 뿌리 노드와 노드 A_1은 아무것도 기재되어 있지 않은 서비스 목록을 갖는다. (b) 세션 S_1이 새롭게 저장 기간 내에 있게 될 때 갱신된 서비스 목록들. (c) 세션 S_1 이후에 세션 S_2가 새롭게 저장 기간 내에 있게 될 때 갱신된 서비스 목록들. 그림에서 새롭게 저장되는 모든 노드들은 음영이 지도록 표기하였다.

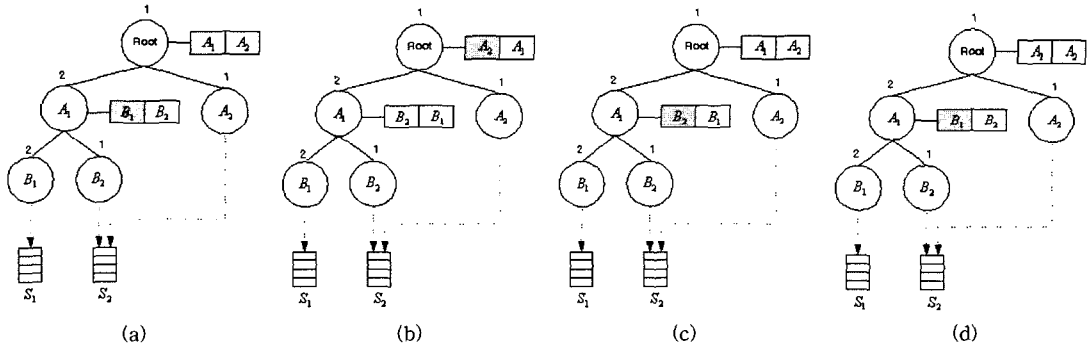


그림 3 갱신된 서비스 목록들. 음영진 항목에 의해서 가르켜지는 잎 노드가 결국 서비스 슬롯을 받는다. (a) 첫 번째와 두 번째 서비스 슬롯이 주어질 때. (b) 세 번째 서비스 슬롯이 주어질 때. (c) 네 번째 서비스 슬롯이 주어줄 때. (d) 다섯 번째 서비스 슬롯이 주어질 때

것은 아니다. 각 잎 노드 k는 이전 서비스 슬롯에서 평균 Q 바이트 전송을 만족시키기 위해서 결손된 양을 기록하는 결손 카운터(deficit counter) d_k를 유지한다. 초기에, 결손 카운터 d_k는 영으로 설정된다. 잎 노드 k가 서비스 슬롯을 받았을 때, 그 세션 큐가 비어 있지 않고 전송된 양이 Q에다가 d_k를 더한 양보다 크지 않을 때까지 그 큐로부터 패킷을 전송한다. 노드 k를 서비스한 후에 만일 그 세션 큐가 비어 있으면, d_k를 영으로 설정한다. 그렇지 않으면, Q에다가 d_k를 더한 값에다가 전송된 양을 뺀 값을 결손 카운터 d_k에 저장한다. 결손 카운터를 통해서 잎 노드는 다음 서비스 시간에 결손양을 보상한다는 점을 주목해야 한다.

결손 보완 계층적 라운드-로빈 알고리즘의 계산 복잡도는 계층의 깊이 비례하지만, 계층 깊이를 몇 단계로 제한해서 상수이므로 결국 계산 복잡도는 상수로 간주할 수 있다. 제안하는 결손 보완 계층적 라운드-로빈 알고리즘의 의사코드는 [18]에 자세히 기록되어 있다.

3. 공평성 분석

한 그룹에 속하는 임의의 두 세션 i와 j가 저장 기간 내에 있는 모든 시간 구간 $(t_1, t_2]$에 대해서 스케쥴링 알고리즘이 다음의 식을 만족하면 우리는 이 알고리즘이 그룹에 속한 세션들에게 공평하다고 정의한다.

$$\left| \frac{W_i(t_1, t_2)}{r_i} - \frac{W_j(t_1, t_2)}{r_j} \right| \leq B \quad (1)$$

여기서 r_i와 r_j는 각각 세션 i와 j에게 할당된 서비스율(service rate)이고, B는 이 스케줄링 알고리즘의 공평도라고 불리는 상수값이다. B의 값이 더 작을 수록 알고리즘이 더 높은 공평도를 갖게 된다. 참고문헌 [19]에 제안된 공평성 정의는 오직 하나의 그룹이 존재할 때의 특별한 경우가 된다.

결손 보완 계층적 라운드-로빈 알고리즘은 각 레벨에 잎이 아닌 노드를 오직 하나만 갖는 뒤틀린 계층(skewed hierarchy)을 구성함으로써 할당 비율에 기반한 그룹을 형성할 수 있다. 예들 들어 그림 4를 살펴 보자. 이 그림에서 계층 구조는 각 단계에 가중치 10을 갖는 하나의 잎이 아닌 노드, 예들 들어 단계 1의 노드 A_n만을 가지며, $A_1, \cdots, A_{n-1}$은 잎 노드들이다. 각 단계에서의 가중치 합은 100을 넘지 않게 제한된다. 이 경우 가중치 w를 갖는 노드는 자신의 부모 노드에게 주어진 대역폭의 w 퍼센트를 보장받는다. 우리는 각 단계 l에 위치한 잎 노드들에 의해 가르켜지는 세션들의 집합을 g_l 그룹으로 정의한다. 이러한 그루핑(grouping)은 할당 비율에 기반한 그룹을 낳는다. 예들 들어, 전송 링크 대역폭이 100 Mbps이고, 그림 4에 기반한 계층 구조를 사용한다고 가정하자. 만일 한 세션이 10 Mbps를 요구한다면, 이 세션은 단계 1의 잎 노드가 할당되기 때문에 g_1 그룹의 구성원 된다. 만일 세션이 대역폭을 1 Mbps에서 9 Mbps 사이의 대역폭을 요구한다면, 그룹 g_2의 구성원이 된다. 만일 결손 보완 계층적 라운드-로빈 알고리즘이 사용하는 계층의 깊이를 1로 제한한다면, 오직 하나의 그룹만 존재하게 되고, 이 경우가 DRR[14] 알고리즘과 동일해 진다.

결손 보완 계층적 라운드-로빈 알고리즘은 각 할당

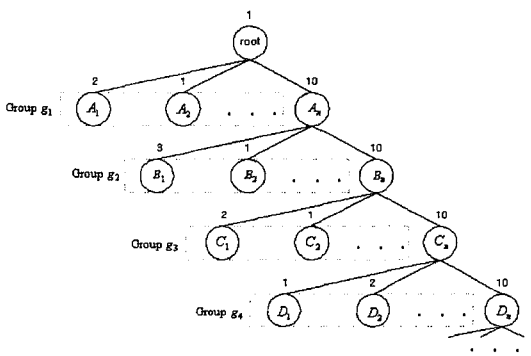


그림 4 할당 비율에 기반한 그룹 형성을 위한 뒤틀린 계층의 예

비율에 기반한 그룹에 속한 세션들에게 공평하다. 이 공평성을 보이기 위해, 우선 저장 기간 동안 잎 노드에 의해서 전송된 패킷의 최소량과 최대량을 추출한다. 그리고 나서, 형제 잎 노드들간에 정규화된 서비스양(normalized service amount)의 차이가 상수값 이내로 한정됨을 보인다. 각 그룹에 속한 모든 세션들은 형제 잎 노드들에게 할당되기 때문에 결손 보완 계층적 라운드-로빈 알고리즘은 각 그룹에 속하는 세션들에게 공평하다.

보조정리 1 결손 보완 계층적 라운드-로빈 알고리즘에서 연속적으로 저장 기간 내에 있는 잎 노드 n을 고려하자. 노드 n의 가중치는 w_n이고, 최대 패킷 크기는 $L^{\max}$이다. 노드 n과 형제 노드들의 k-번째 서비스 라운드의 종료 시간을 r_k로 표기하자. 또한, 첫 번째 라운드의 시작 시간을 r_0으로 표기하자. 만일 K번째까지 서비스 라운드 동안, 즉, 시간 구간 $(r_0, r_K]$ 동안 노드 n이 가르키는 세션 큐에서 전송된 패킷 양을 $W_n(r_0, r_K)$으로 표기하면,

$$Kw_nQ - L^{\max} \leq W_n(r_0, r_K) \leq Kw_nQ + L^{\max}. \quad (2)$$

증명. k번째 서비스 라운드를 끝냈을 때 노드 n의 결손 카운터 값을 D_n^k로 표기하자. k번째 라운드 동안, 잎 노드 n이 가르키는 세션 큐로부터 전송된 패킷양은 $(w_nQ + D_n^{k-1} - D_n^k)$이다. 따라서, 만일 첫 번째 라우드부터 K번째 라운드까지 그 세션 큐로부터 전송된 패킷 양을 합하면,

$$W_n(r_0, r_K) = Kw_nQ + D_n^0 - D_n^K. \quad (3)$$

모든 결손 카운터 값들은 항상 0보다 같거나 크고 최대 패킷 크기 $L^{\max}$보다 작다. 즉,

$$D_n^0 \geq 0 \text{ and } D_n^K < L^{\max}. \quad (4)$$

식 (3)과 식 (4)로부터 이 정리는 증명된다. □

정리 1 결손 보완 계층적 라운드-로빈 알고리즘에서 단계 l에 위치한 잎 노드 m과 n을 고려 하자. 이 두 노드는 형제 노드 관계이고 각각 가중치 w_m과 w_n을 갖고 있다. 또, 이 두 노드를 포함한 형제 노드들의 가중치 합을 w_{sum}으로 표기하자. 노드 m과 n에 각각 할당된 서비스율을 r_m과 r_n으로 표기하고, 최대 패킷 크기를 $L^{\max}$로 표기하자. 그러면, 두 노드 m과 n이 저장 기간 내에 있는 모든 시간 구간 $(t_1, t_2]$를 고려할 때 다음 식을 만족한다.

$$\left| \frac{W_m(t_1, t_2)}{r_m} - \frac{W_n(t_1, t_2)}{r_n} \right| \leq w_{\text{sum}} \left(\frac{L^{\max}}{w_m} + \frac{L^{\max}}{w_n} + Q \right). \quad (5)$$

증명. 시간 구간 $(t_1, t_2]$ 동안 노드 n이 가르키는 세션 큐에서 전송된 패킷 양을 $W_n(t_1, t_2)$로 표기하자. 결손 보완 계층적 라운드-로빈 알고리즘의 변하지 않는

기본 사항 중 하나는, 만일 임의의 시간 구간 $(t_1, t_2]$ 동안 양쪽 노드 m과 n이 저장 기간 내에 있다면, 이 두 노드들 간에 완결된 서비스 라운드 수의 차이는 결코 1을 넘지 않는 다는 것이다. 따라서, 만일 시간 구간 $(t_1, t_2]$ 동안 각 노드 m에게 주어진 완결된 라운드 수를 K_m으로 표기하면, 다음 식을 만족한다.

$$|K_m - K_n| \le 1. \tag{6}$$

식 (6)은 두 가지 경우, $K_m \ge K_n$인 경우와 $K_m \le K_n$인 경우로 나눌 수 있다.

경우 1: $K_m \ge K_n$. 경우 1은 다시 두 가지 경우, 노드 m이 시간 t_2 이전에 이미 (K_m+1)번째 라운드 서비스를 받기 시작한 경우와 그렇지 않은 경우로 나눌 수 있다. 우선 후자의 경우를 먼저 고려 하자. 이 경우 $W_n(t_1, t_2)$는 K_m 라운드 동안 전송된 패킷의 양이 된다. 따라서, 보조정리 1로부터

$$W_m(t_1, t_2) \le K_m w_m Q + L^{\max}. \tag{7}$$

노드 m에 대한 서비스율은 다음 식과 같다.

$$r_m = \frac{w_m}{w_{\text{sum}}}. \tag{8}$$

식 (7)와 식 (8)로부터

$$\frac{W_m(t_1, t_2)}{r_m} \le w_{\text{sum}}\left(K_m Q + \frac{L^{\max}}{w_m}\right). \tag{9}$$

게다가, $K_m \ge K_n$이고 $|K_m - K_n| \le 1$이기 때문에,

$$K_n \ge K_m - 1. \tag{10}$$

따라서, 보조정리 1로부터,

$$\begin{aligned} W_n(t_1, t_2) &\ge K_n w_n Q - L^{\max} \\ &\ge (K_m - 1) w_n Q - L^{\max}. \end{aligned} \tag{11}$$

노드 n에 대한 서비스율은 다음 식과 같다.

$$r_n = \frac{w_n}{w_{\text{sum}}}. \tag{12}$$

식 (11)와 식 (12)로부터,

$$\frac{W_n(t_1, t_2)}{r_n} \ge w_{\text{sum}}\left((K_m - 1)Q - \frac{L^{\max}}{w_n}\right). \tag{13}$$

식 (9)와 식 (13)로부터,

$$\left|\frac{W_m(t_1, t_2)}{r_m} - \frac{W_n(t_1, t_2)}{r_n}\right| \le w_{\text{sum}}\left(\frac{L^{\max}}{w_m} + \frac{L^{\max}}{w_n} + Q\right). \tag{14}$$

이제 전자의 경우, 즉, 노드 m이 시간 t_2 이전에 이미 (K_m+1)번째 라운드 로빈 서비스를 받기 시작한 경우를 고려해 보자. 이 경우 K_n은 K_m은 같다. 만일 그렇지 않다면, 식 (6)은 성립하지 않기 때문이다. 노드 m이 (K_m+1)번째 라운드에 대한 서비스를 받기 시작했기 때문에, 보조정리 1로부터,

$$W_m(t_1, t_2) \le (K_m + 1) w_m Q + L^{\max} \tag{15}$$

이고,

$$\begin{aligned} W_n(t_1, t_2) &\ge K_n w_n Q - L^{\max} \\ &= K_m w_n Q - L^{\max}. \end{aligned} \tag{16}$$

식 (8)와 식 (15)로부터,

$$\frac{W_m(t_1, t_2)}{r_m} \le w_{\text{sum}}\left((K_m + 1)Q + \frac{L^{\max}}{w_m}\right). \tag{17}$$

식 (12)와 식 (16)로부터,

$$\frac{W_n(t_1, t_2)}{r_n} \ge w_{\text{sum}}\left(K_m Q - \frac{L^{\max}}{w_n}\right). \tag{18}$$

식 (17)와 식 (18)로부터,

$$\left|\frac{W_m(t_1, t_2)}{r_m} - \frac{W_n(t_1, t_2)}{r_n}\right| \le w_{\text{sum}}\left(\frac{L^{\max}}{w_m} + \frac{L^{\max}}{w_n} + Q\right). \tag{19}$$

경우 2: $K_m \le K_n$. 증명은 경우 1에서 m과 n을 서로 바꾼 것과 동일하다. 따라서, 결과적으로 다음 식을 얻는다.

$$\left|\frac{W_n(t_1, t_2)}{r_n} - \frac{W_m(t_1, t_2)}{r_m}\right| \le w_{\text{sum}}\left(\frac{L^{\max}}{w_n} + \frac{L^{\max}}{w_m} + Q\right). \tag{20}$$

□

정리 1은 결손 보완 계층적 라운드-로빈 알고리즘이 사용하는 계층의 각각의 단계에서 가중치 합이 작을 수록 높은 공평도를 성취함을 의미한다. 다행히도 결손 보완 계층적 라운드-로빈 알고리즘은 각 단계에서 가중치 값의 합이 작기 때문에 각 그룹에 속한 세션들에게 높은 공평성을 제공할 수 있다. 그 주된 이유는 비슷한 서비스율을 갖는 세션들이 함께 그루핑되기 때문에, 서비스율에 관계없이 작은 가중치 값이 할당되기 때문이다. 즉, 가중치 1이 그룹에 따라서 10 Mbps를 의미할 수도 있고, 1 Kbps를 의미할 수도 있다. 대조적으로, DRR은 세션에 대한 서비스율에 비례해서 가중치가 할당되기 때문에 높은 공평성을 성취할 수 없다. 예를 들어, 만일 1 Kbps 세션에게 가중치 1을 할당한다면, 10 Mbps 세션은 가중치 10000이 할당되어야 한다. 이 때, 가중치 합은 너무 큰 값을 가지게 되어 공평도가 떨어지는 결과를 낳는다.

그러나, 결손 보완 계층적 라운드-로빈 알고리즘은 계층의 깊이가 1인 경우를 제외했을 때, 모든 세션에 대해서 공평한 서비스를 제공할 수는 없다. 그 근본적인 이유를 예를 들어 살펴보자. 노드 p가 두 자식 노드 A_1과 A_2를 갖는다고 가정하자. 노드 A_1은 다시 자식노드로 두 개의 잎 노드 B_1과 B_2를 갖는다. 또한, 노드 A_2도 마찬가지로 두 개의 잎 노드 B_3과 B_4를 갖는다. 이 잎 노드들이 모두 동일한 가중치 1이 할당되었다고 하자. 또, B_3을 제외한 모든 노드가 시간 구간 $(t_1, t_2]$ 동안 저장 기간 내에 있다고 가정하자. 그러면, 노드 B_3이 저장 기간 내에 있지 않기 때문에, 노드 B_4에 의해 전송된 패킷의 양은 노드 B_1이나 노드 B_2

에 의해서 전송된 패킷 양의 두 배가 된다. 이 경우 노드 B_4와 B_1 (혹은 B_2)간의 공평도는 한정되지 않는다. 즉, 만일 시간 구간 $(t_1, t_2]$의 길이가 커지면, 노드 B_4에 의해서 전송된 패킷양은 노드 B_1이나 노드 B_2에 의해 전송된 패킷양보다 얼마든지 클 수 있다.

4. 실험 결과

대역폭은 측정하는 시간 구간에 대해서 전송된 비트 수를 의미한다. 공평성이 높다는 것은 좁은 시간 구간에 대해서도 대역폭이 골고루 분배됨을 의미한다. 이 절에서는 실험을 통해서 결손 보완 계층적 라운드-로빈 알고리즘이 좁은 시간 구간에 대해서도 대역폭을 골고루 분배하며, WFQ 알고리즘에 필적하는 공평성을 제공하는 반면 DRR 알고리즘은 넓은 시간 구간에 대해서도 대역폭이 한 세션에 편중되어 현저히 낮은 공평성만을 제공함을 시뮬레이션을 통해 보인다.

각 알고리즘의 공평성 비교를 위한 실험 환경을 살펴보자. 링크 대역폭은 100 Mbps이고 최대 패킷 크기는 1024 바이트이다. 세션은 s1부터 s15까지 15개를 고려한다. 세션 s5, s10, s15는 각각 20 Mbps, 2 Mbps, 200 Kbps로 데이타를 전송하는 CBR(Constant-Bit-Rate) 세션인데 500ms 간격으로 데이타 전송 및 중단을 반복한다. 나머지 세션들은 네트워크를 포화시킬 만큼의 충분한 양의 트래픽을 보낸다. 이 때 세션 s1, s2, s3은 20 Mbps, s4는 10 Mbps, s6, s7, s8은 2 Mbps, s9는 1 Mbps, s11, s12, s13는 200 Kbps, s14는 100 Kbps 이상의 대역폭을 보장받기를 원한다. 이 경우 결손 보완 계층적 라운드-로빈 알고리즘은 Q 값을 최대 패킷 크기인 1024로 설정하고 각 계층의 가중치 합은 최대 10으로 제한한다. 그림 5는 이 예에서 결손 보완 계층적 라운드-로빈 알고리즘이 사용하는 계층 구조와 각 세션이 필요로 하는 대역폭을 보장하기 위해서 할당된 가중치 값을 보여준다. DRR 알고리즘의 경우는 가중치 1을 갖는 세션이 100 Kbps를 보장 받도록 설정한다. 그러면, 1 Mbps와 10 Mbps 대역폭을 원하는 세션 s9와 s4의 경우 각각 가중치 10과 100을 할당 받는다.

그림 6과 그림 7은 각각 WFQ와 결손 보완 계층적 라운드-로빈 알고리즘에 의해서 10 ms 구간 간격으로 세션 s1, s2, s3, s4에게 제공된 대역폭을 보여준다. 두 그림에서 대역폭이 각 세션들에게 골고루 분배됨을 볼 수 있다. 반면 그림 8은 DRR 알고리즘의 경우 대역폭 측정 구간을 50 ms 간격으로 넓혀서 측정해도 각 세션들에게 대역폭이 골고루 분배되지 않고 편중됨을 알 수 있다.

또, 세션 s6, s7, s8, s9에 제공된 대역폭을 20 ms 구

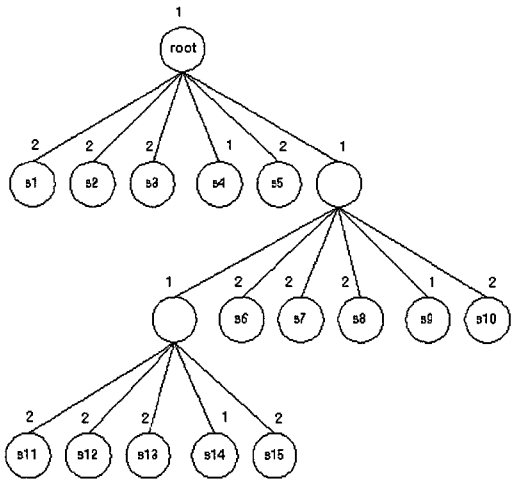


그림 5 실험에 사용된 계층구조

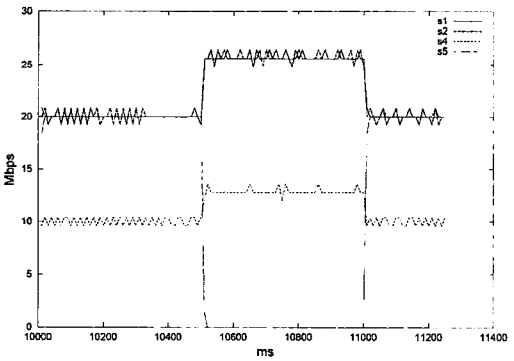


그림 6 10 ms 구간 간격으로 측정했을 때 WFQ 알고리즘이 세션 s1, s2, s3, s4에게 제공한 대역폭

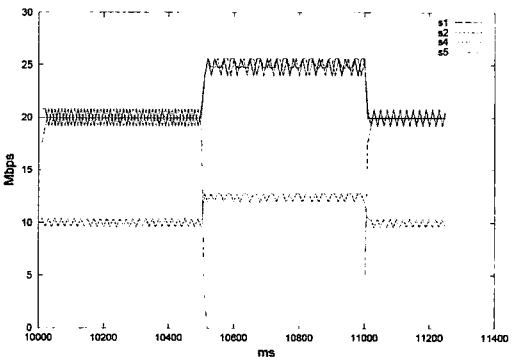


그림 7 10 ms 구간 간격으로 측정했을 때 결손 보완 계층적 라운드-로빈 알고리즘이 세션 s1, s2, s3, s4에게 제공한 대역폭

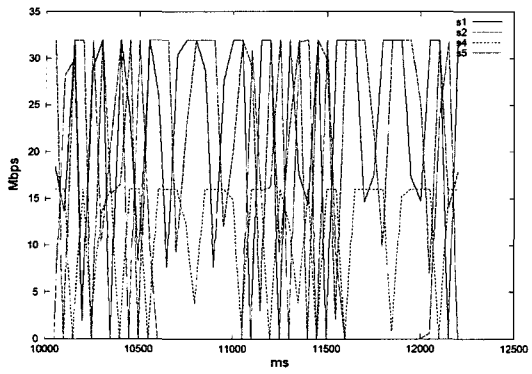


그림 8 DRR 알고리즘이 세션 s1, s2, s3, s4에 대해서 구간 50 ms구간 간격으로 측정했을 때 제공한 대역폭

간 간격으로 측정했을 때, WFQ와 결손 보완 계층적 라운드-로빈 알고리즘의 경우 앞의 경우와 비슷하게 대역폭이 세션들에게 골고루 분배됨을 확인할 수 있었고, DRR 알고리즘의 경우는 각 세션들에게 대역폭이 골고루 분배되지 않고 편중됨을 알 수 있었다. 이 경우의 결과 그래프는 앞의 경우와 거의 동일하기 때문에 생략하였다.

5. 결 론

이 논문은 상당한 수준의 공평성을 제공하는 결손 보완 계층적 라운드-로빈 알고리즘을 제안한다. 즉, 좁은 시간 구간에 대해서 대역폭을 측정했을 때 같은 그룹에 속한 세션들에게 대역폭을 골고루 분배한다. 큰 시간 주기가 아니라 작은 구간에 대해서 대역폭을 골고루 분배하는 것은 TCP/IP 프로토콜을 사용하는 응용 프로그램의 성능이 공평한 결과를 낳는다는 측면에서 매우 중요하다[20]. 역으로 큰 시간 구간에 대해서 대역폭이 골고루 분배된다 할지라도 TCP/IP를 사용하는 응용들이 동일한 성능을 갖지 않을 수 있음을 의미한다. 본 연구에서 제안한 결손 보완 계층적 라운드-로빈 알고리즘은 이와 같은 장점을 가지고 있으면서도 구현 복잡도가 상수 시간이기 때문에 확장성을 갖는다.

참 고 문 헌

[1] A. K. J. Parekh and R. G. Gallager. A Generalized Processor Sharing Approach to Flow Control in Integrated Service Networks: The Single Node Case. *IEEE/ACM Tran. Networking*, 1(3):344-357, June 1993.

[2] A. K. J. Parekh and R. G. Gallager. A Generalized Processor Sharing Approach to Flow Control in Integrated Service Networks: The Multiple Node Case. *IEEE/ACM Tran. Networking*, 2(2):137-150, April 1994.

[3] J.C.R. Bennett and H. Zhang. Hierarchical Packet Fair Queueing Algorithms. *IEEE/ACM Tran. Networking*, 5(5):675-689, October 1997.

[4] Pawan Goyal, Harrick M. Vin, and Haichen Cheng. Start-Time Fair Queueing: A Scheduling Algorithm for Integrated Services Packet Switching Networks. *IEEE/ACM Tran. Networking*, 5(5):690-704, October 1997.

[5] D. Stiliadis and A. Varma. Rate-Proportional Servers: A Design Methodology for Fair Queueing Algorithms. *IEEE/ACM Tran. Networking*, 6(2): 164-174, April 1998.

[6] Jeng Farn Lee, Yeali Sun, and Meng Chan Chen. On Maximum Rate Control of Weighted Fair Scheduling for Transactional Systems. In *Real-Time Systems Symposium*, pages 335-344, 2003.

[7] G. Kornaros, T. Orphanoudakis, and I. Papaefstathiou. GFS: An Efficient Implementation of Fair Scheduling for Multigigabit Packet Networks. In *IEEE International Conference on Application-Specific Systems, Architectures, and Processors*, pages 389-399, 2003.

[8] J. H. Anderson, A. Block, and A. Srinivasan. Quick-Release Fair Scheduling. In *Real-Time Systems Symposium*, pages 130-141, 2003.

[9] Hanrijanto Sariowan, Rene L. Cruz, and George C. Polyzos. SCED: A Generalized Scheduling Policy for Guaranteeing Quality-of-Service. *IEEE/ACM Tran. Networking*, 7(5):669-684, October 1999.

[10] Ion Stoica, Hui Zhang, and T. S. Eugene Ng. A Hierarchical Fair Service Curve Algorithm for Link-Sharing, Real-Time and Priority Services. *IEEE/ACM Tran. Networking*, 8(2):185-199, 2000.

[11] Norival R. Figueira and Joseph Pasquale. A Schedulability Condition for Deadline-Ordered Service Disciplines. *IEEE/ACM Tran. Networking*, 5(2):232-244, April 1997.

[12] Massoud R. Hashemi and Alberto Leon-Garcia. The Single-Queue Switch: A Building Block for Switches with Programmable Scheduling. *IEEE JSAC*, 15(5):785(794, June 1997.

[13] P. Lavoie and Y. Savaria. A systolic architecture for fast stack sequential decoders. *IEEE Tran. Communications*, 42(2/3/4):324-335, Feb./Mar./Apr. 1994.

[14] M. Shreedhar and George Varghese. Efficient Fair Queuing Using Deficit Round-Robin. *IEEE/ACM Tran. Networking*, 4(3):375-385, June 1996.

[15] C. R. Kalmanek, H. Kanakia, and S. Keshav. Rate Controlled Servers for Very High-Speed Networks. In *IEEE Global Telecommunications Conference*, pages 12-20, 1990.

[16] Hemant M. Chaskar and U. Madhow. Fair scheduling with tunable latency: A Round Robin

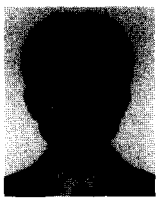
approach. *IEEE/ACM Tran. Networking*, 11(4): 592-601, August 2003.

[17] Salil S.Kanhere and Harish Sethu. Fair, Efficient and Low-Latency Packet Scheduling Using Nested Deficit Round Robin. In *IEEE Workshop on High Performance Switching and Routing*, pages 6-10, 2001.

[18] Kihyun Pyun. *Packet Scheduling Algorithms to Provide Real-Time, Fair, and Link-Sharing Services in Integrated Services Networks*. PhD thesis, Department of Electrical Engineering & Computer Science, Division of Computer Science, KAIST, February 2003.

[19] S. Golestani. A Self-Clocked Fair Queueing Scheme for Broadband Applications. In *INFOCOM*, pages 636-646, 1994.

[20] A. Demers, S. Keshav, and S. Shenkeer. Analysis and Simulation of a Fair Queueing Algorithm. In *ACM SIGCOMM*, pages 3-12, 1989.



편 기 현

1995년 인하대학교 전자계산공학과(학사) 1997년 KAIST 전산학과(석사). 2002년 KAIST 전산학과(박사). 2002년~2003년 KAIST 전기및전자공학, 박사후 연구원(Post Doctor). 2004년~현재 전북대학교 전자정보공학부 전임강사. 관심분야는 유무선 고품질 서비스, 유무선 패킷 스케줄링, 자원 관리, 저전력 시스템 소프트웨어

조 성 익

1987년 전북대학교 전기공학과 학사 졸업. 1989년 전북대학교 전기공학과 석사 졸업. 1994년 전북대학교 전기공학과 박사 졸업. 1994년~2004년 Hynix 반도체 메모리 연구소 책임연구원. 2004년~현재 전북대학교 전자정보공학부 교수. 관심분야는 저전압/고속 Graphic DRAM, Low-voltage Low-power analog circuit, High speed data Interface circuit, ADC/DAC, PLL/DLL>

이 종 열

1993년 2월 한국과학기술원 전자전산학과(공학사). 1996년 2월 한국과학기술원 전자전산학과(공학석사). 2002년 8월 한국과학기술원 전자전산학과(공학박사). 2002년 9월~2003년 9월 하이닉스 반도체 선임연구원. 2003년 10월~2004년 2월 한국과학기술원 BK21 초빙교수. 2004년 3월~현재 전북대학교 전자정보공학부 전임강사. 관심분야는 SOC 설계, 내장형 프로세서 설계, 내장형 소프트웨어 최적화