Non-independent system에서의 공정성 평가

김학용 · 김기선

공정성은 공유 시스템의 성능 연구에 있어서 중요한 고려 사항의 하나이다. 현재까지는 Jain이 1984년에 정의한 공정성 및 공정성 지수를 사용해서 관측 시스템의 공정성을 정량적으로 평가하고 있다. 그러나 Jain의 정의는 단일 혹은 독립적인 시스템의 공정성만을 평가할 수 있어서, 서로 상관관계가 있는 시스템에서 공정성을 보장하려 할 때는 시스템 자원을 효율적으로 사용하지 못하는 문제점을 야기시킨다. 따라서 본 논문에서는 Jain의 공정성 정의를 변경시킴으로써 입력 큐잉 스위치를 비롯한 독립적이지 않은 시스템에서 적용할 수 있도록 하였다. 새로운 공정성 정의는 기존의 Jain의 공정성 지수가 가지고 있는 모든 특성을 유지하고 있으며, 시스템 자원을 효율적으로 사용할 수 있도록 하고 있다.

주제어 : 공정성, 공정성 지수

I. 서 론

공정성(fairness)은 대부분의 성능 연구에 있어서 중요한 고려 사항의 하나이다. 특히, 여러 사용자들이 시스템의 자원(resource) 혹은 용량(capacity)을 공유하는 분배 시스템(distributed systems)이나 여러 커넥션들이 제한된 수의 서버를 놓고 경쟁하는 경우에 서버 능력의 공정한 할당(fair allocation)은 중요하다. 따라서 공정성에 대한 많은 연구가 이루어져 왔으며 [1],[2],[3],[4],[5], 참고문헌 [5]에서는 다섯 가지의 서로 다른 공정성의 기준들을 각각의 특성과 함께 제시하고 있다.

어떤 시스템의 공정성을 이야기 할 때 공정하다 혹은 그렇지 않다 라고도 할 수 있으나, 이런 정성적인 방법은 그다지 직관적이지 않다. 반면에 그 시스템은 90% 공정하다거나 10% 불공정하다고 정량적으로 이야기 하는 것은 비록 비전문가일지라도 쉽게 시스템의 공정성을 이해하도록 한다. 그럼에도 불구하고, 공정성 및 공정성을 정량적으로(quantitatively) 평가하는 방법에 대해서는 Jain에 의한 연구[1],[2],[3] 이외에는 찾아보기가 힘들 정도이다. Jain에 의한 공정성의 정의는 II장에서도 살펴보겠지만, Jain에 의한 공정성의 정의는 간단하면서도 직관적이고 이론적으로도 타당하므로 통신 시스템을 비롯한 여러 분야에서 공정성을 평가하는데 사용되고 있다. Jain의 공정성에 대한 연구 이외의 다른 공정성에 대한 연구들은 특정 시스템 혹은 특정한 특성(metric)에 한정된 것이거나 정성적인(qualitatively) 경향이 있어서 모든 시스템이나 어플리케이션에 포괄적으로 사용할 수 없다. (이에 대한 리뷰 및 참고문헌을 [1]에서 찾아볼 수 있다.) 따라서, 본 논문에서는 Jain 논문에서 이야기된 공정성만을 생각할 것이다.

참고문헌 [1]에서 Jain은 공정성의 문제를 적절한 할당 특성표(proper allocation metric)를 선택하는 것과, 이와 같은 여러 할당(allocation) 사이의 공정성을 정량적인 값으로 나타낼 수 있는 공식을 정의하는 것으로 나누어서 생각했다. 특성표의 선택이 어플리케이션이나 사용자들의 요구에 의존적이기 때문에, 저자들은 특성표의 선택보다는 공정성에 대한 정의 및 공정성을 정량적으로 나타낼 수 있는 공식(fairness index)에 관한 측면에 논의를 집중하였다. 그 공식은 할당의 공정성을 측정할 수 있으며, 공정하지 않은 경우 얼마나 공정하지 않은지를 나타낼 수 있는 것이다. 또한, 공정성

김학용 : 코어세스
김기선 : 광주과학기술원

지수(fairness index)의 바람직한 특성에 대해서 논의하였다. 이 내용은 II장에서 간단히 검토될 것이다.

Jain에 의해 정의된 공정성 및 공정성 지수는 출력 큐잉 방식의 패킷 스위치에 대한 스케쥴링 알고리즘을 평가하는 데에도 사용되고 있다[4]. 출력 큐잉 스위치에서는 모든 패킷들이 스위치에 도착하자마자 큐잉 혹은 큐잉에 의한 지연 없이 출력포트로 보내어진다. 각 출력포트에서 패킷들은 특정 시간 동안 제한된 수의 패킷을 서비스해 줄 수 있는 서버를 놓고 경쟁해야만 한다. 이런 출력 큐잉 스위치는 일종의 독립적인 공유 시스템으로 생각할 수 있기 때문에, Jain에 의해 정의된 공정성 및 fairness index를 아무런 수정 없이 사용할 수 있었다.

그러나 고속 스위칭에 대한 요구가 증가함에 따라 패킷 스위치의 구조가 입력 큐잉의 구조로 바뀌고 있다. 입력 큐잉 스위치에서는 출력포트에 메모리가 없으므로 여러 커넥션들이 입력포트를 공유하는 시스템으로 생각할 수가 있다. 그러나 출력 큐잉 스위치와는 달리 특정 입력포트 내에 존재하는 커넥션들은 서비스를 받기 위해 다른 입력포트에 존재하는 목적지가 같은 커넥션들과도 경쟁을 해야 한다. 이는 패킷 서비스의 측면에서 임의의 입력 포트가 다른 입력포트에 대해 독립적이지 않음을 의미하며, 따라서 기존의 Jain에 의한 정의를 그대로 사용할 수 없음을 의미하는 것이다. 만약, Jain의 정의를 그대로 사용한다면, 특정 입력포트에서의 지역적인 공정성(local fairness)은 보장할 수 있을지 모르지만, 스위치 전체의 공정성(global fairness)은 보장할 수 없게 된다. 또한, 전체 시스템의 자원을 효율적으로 사용할 수 없을 뿐만 아니라 다른 성능 특성들에도 영향을 미치게 된다. 따라서 본 논문에서는 Jain의 정의를 변형시킴으로써 입력 큐잉 스위치처럼 독립적이지 않은 공유 시스템에서의 공정성을 정량적으로 평가할 수 있는 방법에 대해 고찰할 것이다. 제안된 방법은 시스템의 공정성 뿐만 아니라 시스템 자원의 효율성을 동시에 보장해 준다.

본 논문의 구성은 다음과 같다. II장에서는 Jain이 정의한 공정성과 공정성 지수 및 공정성 지수가 가져야 할 바람직한 특성들을 살펴볼 것이다. III장에서는 Jain의 정의가 입력 큐잉 혹은 입출력 큐잉 스위치처럼 독립적이지 못한 시스템에서 공정성을 평가할 때 발생할 수 있는 문제점들을 지적할 것이다. IV장에서는 Jain의 정의를 독립적이지 못한 시스템에서도 효율적으로 적용할 수 있도록 utilization의 측면에서 공정성 지수를 새롭게 정의를 할 것이며, 몇몇 예를 통해 그 정당성을 보일 것이다.

II. Jain의 fairness 및 fairness index

Jain은 개체 i의 정규자원할당(normalized resource allocation) x_i를 비교함으로써 어떤 시스템의 공정성을 평가하는 방법을 제안했다[1],[2],[3]. 이때, x_i는 실제 측정된 throughput(y_i)의 이상적인 throughput(z_i)에 대한 비율($x_i = y_i/z_i$)이다. 즉, x_i들이 모두 1.0의 값을 가지고 있으면 그 시스템은 공정하다고 이야기할 수 있으며[2],[3], x_i가 서로 다른 값을 가지고 있으면 불공정하다고 이야기를 했다.

그러나 x_i들을 비교하는 자체로는 정량적으로 공정성을 평가할 수 없으므로 공정성 지수를 정의하였으며 다음과 같다:

$$\text{fairness index} = \frac{(\sum x_i)^2}{n \times \sum x_i^2}. \qquad (1)$$

식 (1)에서 n은 고려 대상이 되는 개체들의 수를 나타낸다. 이 공정성 지수는 각 개체들에 할당된 시스템 자원의 균등성을 측정할 수 있다. 만약, 모든 개체들이 동일한 할당을 받았다면 즉, 모든 x_i가 같은 값을 갖는다면, 공정성 지수는 1.0이 될 것이며 이는 그 시스템은 100% 공정함을 의미하는 것이다. x_i 값들의 차이가 커질수록(분산이 커질수록) 공정성 지수는 감소하게 되며 이는 시스템이 불공정해져 가는 추이를 보여준다. 공정성 지수가 0.0이 되면 완전히 불공정한 시스템을 의미한다. 식 (1)에 보이는 공정성 지수는 분자 및 분모를 n^2으로 나눔으로써 평균과 분산을 사용해서 나타낼 수도 있다.

$$\text{fairness index} = \frac{(\frac{1}{n}\sum x_i)^2}{\frac{1}{n} \times \sum x_i^2} = \frac{(E[X])^2}{E[X^2]}$$

$$= \frac{(E[X])^2}{(E[X])^2 + \text{var}(X)}. \qquad (2)$$

또한, 식 (2)는 분자 분모를 $(E[X])^2$으로 나눔으로써 분산계수(coefficient of variance:C_X)로 나타낼 수도 있다.

$$\text{fairness index} = \frac{1}{1 + \text{var}(X)/(E[X])^2} = \frac{1}{1 + C_X^2} \qquad (3)$$

뒤에서 살펴보겠지만, 식 (2)는 var(X)가 랜덤 변수 X의 단위에 따라 그 값이 변하는 특성을 가지고 있고 식 (3)에서는 C_X는 그 값이 무계(unbounded)인 특성을 가지고 있으며, 두 가지 모두 공정성 지수와 비선형적인 역

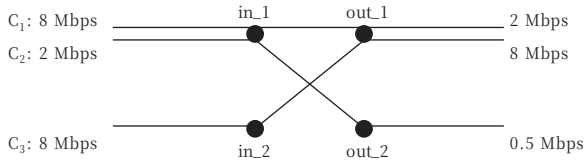

그림 1. 2×2 입력 큐잉 스위치에서 예: in_1에서 두 커넥션 사이의 공정성을 유지하려고 하면 커넥션 C_2는 2 Mbps로 서비스 될 수 있음에도 불구하고 0.5 Mbps로 밖에 서비스 될 수 없다.

비례(nonlinear reciprocal proportion)의 관계가 있으므로 직관적이지 않다. 따라서 식 (2)나 (3)의 표현보다는 식 (1)의 표현이 가장 이해하기 쉽다.

Jain은 공정성과 반대되는 개념으로 차별성(discrimination) 및 차별성 지수(discrimination index)를 정의하였다. 즉, 차별성 지수는 다음과 같이 정의된다:

$$\text{Discrimination index} = 1 - \text{fairness index} = 1 - \frac{(\sum x_i)^2}{n \times \sum x_i^2}.$$

차별성 및 차별성 지수는 공정성 및 공정성 지수로부터 직접적으로 유추될 수 있으므로, 본 논문에서는 차별성에 대한 논의는 하지 않을 것이다.

[1],[2],[3]에서 Jain은 fairness index가 가져야 하는 바람직한 다섯 가지 특성을 제시하고 있으며 다음과 같다.

- *Population size independency*: 공정성 지수는 어떤 수의 개체에도 적용할 수 있어야 한다.
- *Scale and metric independency*: 공정성 지수는 할당의 단위나 크기에 무관해야 한다.
- *Boundedness*: 공정성 지수는 그 값이 유한해야 하며, 가능하면 0과 1사이로 한정되야 한다.
- *Continuity*: 공정성 지수는 연속적인 값을 가져야 한다.
- *Intuitiveness*: 공정성 지수는 정량적이어야 하며 공정성과 직접적인 혹은 선형적인 관계를 가지고 있어야 한다.

Fairness index의 다섯 가지 특성에 대한 보다 자세한 논의 및 관련 예제들은 참고문헌[1],[2],[3]을 참조하기 바란다. 식 (1)에 보이는 fairness index는 이 모든 특성을 만족하고 있음을 쉽게 발견할 수가 있다.

Ⅲ. Jain의 공정성 및 공정성 지수의 문제점

II장에서 살펴본 Jain의 공정성에 대한 정의 및 fairness index는 여러 사용자들이 시스템의 자원(resource) 혹은 용량(capacity)을 공유하는 분배 시스템(distributed systems)이나 여러 커넥션들이 제한된 수의 서버를 놓고 경쟁하는 경우에 적절하게 사용될 수 있다. 특별한 언급은 없었지만, Jain은 공유 시스템이 단독으로 존재하는 경우나, 여러 공유 시스템이 존재하는 경우 다른 시스템들에 대해 독립적인 경우를 가정하고 있다. 즉, 하나의 특정 공유 시스템 내에 있는 개체들은 다른 시스템에 있는 개체들에 대한 자원 혹은 서비스의 할당과 무관하다. 이러한 이유로, Jain의 정의는 출력 큐잉 스위치에서 커넥션들의 공정성을 평가할 때도 사용되고 있다[4].

하지만 관찰 대상이 되는 공유 시스템이 혼재하는 다른 시스템에 대해 독립적이지 않은 경우에는 Jain의 정의를 그대로 쓸 수가 없다. 그 이유는 특정 시스템 내에서의 공정성을 보장하는 것이 때때로 전체 시스템의 성능(utilization)을 저하시킬 수도 있기 때문이다. 대표적인 예로, 입력 큐잉 스위치를 생각해 볼 수 있다. 입력 큐잉 스위치는 스위치에 도착한 패킷들이 각 입력포트에 있는 큐에 잠시 저장되었다가 스케쥴링 알고리즘에 의해 결정되는 순서에 따라 패킷들의 해당 목적지로 보내어지는 시스템이다. 같은 입력포트에 존재하는 여러 커넥션들이 각 입력포트의 서비스 능력을 공유함으로 일종의 공유 시스템으로 볼 수 있다. 그러나 어떤 패킷이 특정 목적지로 서비스되기 위해서는 다른 입력포트에서 동일한 목적지로 향하는 패킷들과의 경쟁에서 이겨야만 하며, 따라서 특정 입력포트는 다른 입력포트에 대해 독립적이지 못한 시스템이 된다.

그림 1은 이런 입력 큐잉 스위치의 가장 간단한 예를 보이고 있다. 즉, 2×2 스위치가 있으며, 입력포트 in_1에는 두 개의 커넥션 C1과 C2가 존재하며, 입력포트 in_2에는 하나의 커넥션 C3만이 존재한다. 커넥션들은 각각 8 Mbps, 2 Mbps, 그리고 8 Mbps의 일정한 비트율을 갖는다고 가정하였으며, 각 입력포트 및 출력포트는 10 Mbps의 서비스 능력을 가지고 있다. 입력 큐잉 스위치에서는 출력포트에 큐가 존재하지 않으므로 특정 출력으로 향하는 트래픽의 양이 각 출력포트의 서비스 능력인 10 Mbps를 넘어서는 안 된다. 따라서 그림1에서는 출력포트 out_1으로 향하는 두 커넥션 C1과 C3가 각각 2 Mbps와 8 Mbps를 전송하도록 결정되었다고 가정하였다. 이런 경우, 기존의 Jain에 의한 공정

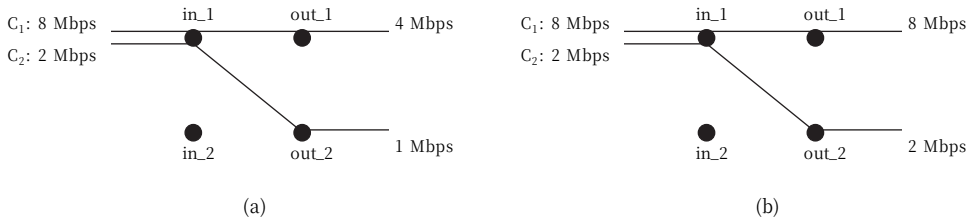

그림 2. 공정한 스위칭 시스템에서의 자원의 활용성 비교

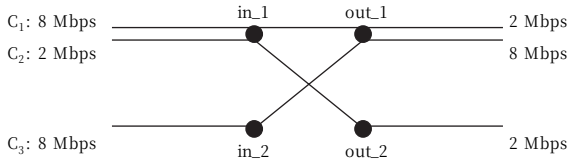

그림 3. 2×2 입력 큐잉 스위치에서 예: in_1에서 두 커넥션 사이의 공정성 보다는 시스템 자원을 최대로 이용하도록 설정되었다. 이 경우 out_2로 향하는 케넥션은 C_2 뿐이므로 C_2를 2 Mbps로 서비스해 줄 수 있다.

성의 정의를 입력포트 in_1에 적용하면 출력포트 out_2로 향하는 커넥션 C2는 C1과의 공정성을 유지하기 위해 0.5 Mbps만을 전송할 수 있게 된다. 출력포트 out_2로 향하는 다른 커넥션이 존재하지 않으므로 2 Mbps의 서비스를 받을 수 있음에도 불구하고 입력포트 in_1에서의 공정성을 유지하기 위해 출력포트 out_2의 서비스 능력을 낭비하게 되는 것이다. 만약, 출력포트에 대해 보다 공정하고 시스템 자원을 최대로 사용하도록 해주는 스케쥴러가 사용되었다면, C1과 C3는 각각 5 Mbps를 전송하도록 결정될 것이다. 이런 경우에는 C2는 C1과의 공정성을 위해 1.25 Mbps를 전송할 수 있게 되지만, 여전히 2 Mbps로 전송하지 못함으로써 출력포트 out_2의 서비스 능력을 낭비하게 된다. 출력포트에 대해서 불공정한 스케쥴러가 사용되어 C1과 C3가 각각 8 Mbps 및 2 Mbps를 서비스 하도록 결정되었다면, C2는 2 Mbps로 서비스를 받을 수 있게 되며 출력포트 out_2의 서비스 능력을 낭비하지 않게 된다. 그러나 이런 경우 입력포트 in_1에 있는 커넥션들은 100% 서비스를 받음에도 불구하고 입력포트 in_2에 있는 커넥션은 고작 25%의 서비스만을 받음으로 인해 입력포트 사이의 공정성에 영향을 미치게 된다.

Jain의 공정성 및 공정성 지수의 또 다른 문제점은 전체 시스템의 효율성을 고려하지 않고 있다는 점이다. 그림 2에 보이는 예를 살펴보자. 그림 2 (a)의 경우 x1 및 x2는 모두 0.5의 값을 갖게 되며, 따라서 공정성 지수는 1.0이 되어 공정한 시스템을 나타낸다. 그림 2 (b)의 경우는 x1 및 x2가 모두 1.0이 되고 공정성 지수도 1.0이 되어 공정한 시스템을 나타내고 있다. 그러나 (b)의 경우는 두 커넥션 C1 및 C2에 대해 100% 서비스를 제공하고 있지만, (a)의 경우는 50%의 서비스만을 제공하고 있다. Jain은 (b)의 경우처럼 모든 xi가 1.0과 같도록 자원을 할당하는 시스템을 이상적으로 공정한 시스템이라고 이야기 하고 있으나, Jain이 정의한 공정성 지수로는 두 가지 경우를 구분할 수 없게 된다. IV장에서는 Jain의 공정성에 대한 정의의 문제점들을 해결하기 위해, 공정성을 utilization 측면에서 새로이 정의하였다.

IV. Utilization - Wise Fairness Definition

III장에서는 [1],[2],[3]에서 소개된 Jain의 공정성에 대한 정의가 시스템 자원의 효율성이라는 측면에서 갖는 두 가지 문제점을 고찰하였다. 두번째로 지적되었던 문제는 실제 공유 시스템은 자원을 최대로 활용할 수 있도록 설계됨으로 시스템 자원의 비효율적인 할당 문제는 중요하지 않게 된다. 문제는 독립적이지 않은 공유 시스템에서 개체들에게 시스템 자원을 공정하게 할당하는 경우에 발생하는 문제점이다. 즉, 모든 개체에 각 개체가 요구하는 자원에 비례하도록 자원을 할당함으로써 시스템 전체의 utilization을 저하시키는 것이다. 이를 해결하기 위해서는 시스템에서 지원할 수 있는 범위 내에서 자원을 최대로 할당하는 것이다. 그러나 이렇게 되면, 기존의 Jain의 공정성에 대한 정의는 더 이상 사용할 수 없게 된다. 그림 3은 이러한 문제를

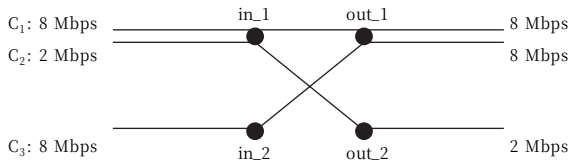

그림 4. Internal speedup factor가 2인 입출력 큐잉 스위치에서의 예

설명하고 있다. 그림 3은 그림 1에서 0.5 Mbps만 서비스 받을 수 있었던 커넥션 C2가 2 Mbps 만큼 서비스되도록 자원을 할당한 경우이다. 이런 경우에 Jain의 정의를 사용하면, x_1 및 x_2는 각각 0.25와 1.0의 값을 갖게 되어 입력포트 in_1에서의 공정성 지수는 0.735가 되어 시스템 자원을 최대로 활용함으로써 36.5% 만큼 불공정한 시스템이 되어 버린다. 그러나 실제로 커넥션 C1은 커넥션 C2가 2 Mbps를 서비스 받아도 자신이 서비스에는 아무런 영향을 주지 않으므로 불만이 없다고 볼 수 있으며, 두 커넥션 사이에는 공평한 서비스가 이루어지고 있다고 할 수 있다. 따라서 이러한 점을 감안하기 위해 새롭게 공정성을 정의할 필요가 있다.

따라서 본 논문에서는 정규 자원 할당 지수 x_i를 각 개체에 할당된 자원의 최소값($\min(z_{i\_in}, z_{i\_out})$)에 대한 스위치의 입력 및 출력에서 실제 측정된 throughput y_i의 비율로 정의를 했다. 즉,

$$x_i = \frac{y_i}{\min(z_{i\_in}, z_{i\_out})} \tag{4}$$

가 된다. 공정성 지수는 식 (4)를 식 (1)에 사용함으로써 구할 수 있다. 즉, Jain의 공정성 및 공정성 지수의 형태 및 특징을 그대로 유지하고 있다. 그림 3에 보이는 예에 대해 x_1, x_2 및 공정성 지수는

$$x_1 = \frac{2}{\min(8, 2)} = \frac{2}{2} = 1.0,$$

$$x_2 = \frac{2}{\min(2, 10)} = \frac{2}{2} = 1.0,$$

$$\text{fairness index} = \frac{(2.0)^2}{2\times(1.0+1.0)} = 1.0$$

과 같이 되어 개체들 사이에 공정한 할당을 할 수 있으며, 시스템의 자원을 최대로 활용할 수 있게 된다.

그림 1에서 처럼 고전적인 기준에 의해 C2를 서비스 하는 경우는 시스템 전체의 자원 활용도가 출력포트에 대한 스케쥴링 알고리즘에 따라 달라지지만, 공정성에 대한 새로운 정의를 사용하는 경우는 스케쥴링 알고리즘에 상관없이 시스템 전체의 자원 활용도가 항상 일정한 값을 유지하게 된다. 예를 들면, out_1로 향하는 두 커넥션 C1 및 C3에 대해 2:8, 4:6, 5:5, 6:4 혹은 8:2 등 어떤 비율로 할당을 해도 공정성 지수 및 자원 활용도는 변함이 없다.

새로 정의된 공정성 및 공정성 지수는 입력 큐잉 스위치뿐만 아니라 입출력 큐잉(CIOQ:Combined Input-Output Queueing) 스위치에도 적용될 수 있다. 입출력 큐잉 스위치는 internal speedup으로 인해 입력포트 및 출력포트에 있는 서버의 능력이 speedup factor 만큼 증가하는 것으로 생각할 수 있다. 일반적으로, speedup factor에 의한 내부 동작속도의 증가는 시스템 구현에 있어서 바람직하지 않은 요인으로 인식되고 있으나, speedup factor가 2만 되어도 출력 큐잉 스위치와 동일한 성능을 제공할 수 있다는 최근의 연구결과[8]가 발표됨에 따라, 입출력 큐잉 스위치가 고속 고성능 패킷 스위치 분야에서 다시 주목을 받고 있다. 따라서 본 논문에서도 speedup factor가 2인 경우만을 고려하기로 한다. 그림 4에는 internal speedup factor가 2인 경우의 예를 보여주고 있다. 이 경우 각 입력포트 뿐만 아니라 각 출력포트도 메모리를 가지고 있으므로 각 출력포트는 최대 20 Mbps의 용량으로 입력포트로부터 트래픽을 받을 수 있다. 하지만 입력포트 및 출력포트의 서비스 능력은 여전히 10 Mbps 이므로 그 이상의 트래픽을 동시에 서비스할 수 없게 된다. 따라서 그림 4에 보이는 입출력 큐잉 스위치의 입력포트 in_1에서의 공정성은 다음과 같이 구해진다.

$$x_1 = \frac{8}{\min(8, 10)} = \frac{8}{8} = 1.0,$$

$$x_2 = \frac{2}{\min(2, 20)} = \frac{2}{2} = 1.0,$$

$$\text{fairness index} = \frac{(2.0)^2}{2\times(1.0+1.0)} = 1.0$$

계산된 결과에서도 알 수 있듯이, 입력 큐잉 스위치의

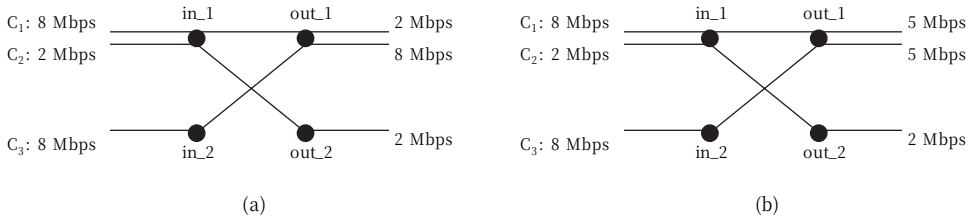

그림 5. 입력포트 및 출력포트 사이의 공정성에 관한 예

경우에서처럼 시스템 자원을 최대한 사용할 수 있도록 하면서 공정성을 유지하고 있음을 알 수 있다.

V. 새로운 정의의 확장

IV장에서 utilization 측면에서 새로이 정의된 공정성 및 공정성 지수는 공유 시스템이 더욱 복잡하게 구성된 경우에도 그대로 적용될 수 있다. 만약 어떤 시스템이 공유 시스템들이 k계층에 걸쳐 구성되었다면, 그 시스템에 대한 정규 자원 할당 지수 x_i는 IV장에서 처럼 각 개체에 할당된 자원의 최소값($\min(z_{i\_1}, z_{i\_2}, z_{i\_3}, \cdots, z_{i\_k})$)에 대한 실제 측정된 throughput y_i의 비율로 정의 할 수 있다. 즉,

$$x_i = \frac{y_i}{\min(z_{i\_1}, z_{i\_2}, z_{i\_3}, \cdots, z_{i\_k})} \quad (5)$$

이며, 이때 z_{i_j}는 모든 j에 대해 0.0 보다 큰 값을 갖아야 한다. $k=1$인 경우는 독립적인 공유 시스템을 의미하며, 이때는 기존의 Jain의 정의와 똑같은 형태를 취함을 알 수 있다. 즉, 출력 버퍼형 스위치에도 그대로 사용할 수 있다.

공정성에 대한 새로운 정의는 시스템이 overload된 경우에도 적용할 수 있다. 우리의 이전 연구 [6]에서는 overload된 경우의 입력 큐잉 스위치에 사용되는 세 가지 스케줄링 알고리즘의 공정성을 비교하였다. 또한, [6]에서 저자들은 다중 입력 큐잉(MIQ:Multiple Input-Queued) 스위치처럼 여러 개체들이 독립적이지 않은 시스템을 공유하는 경우에는 특정 공유 시스템 내부에 속한 개체들의 공정성(local fairness)뿐만 아니라, 입력포트 사이의 공정성 및 출력포트 사이의 공정성처럼 공유 시스템 사이의 공정성(global fairness)도 중요한 문제가 된다고 지적하고 있다. V장에서 새롭게 정의된 공정성 정의는 공유 시스템 사이의 공정성 평가에도 사용할 수 있다. 그림 5는 입력포트 및 출력포트 사이의 공정성을 비교하는 예이다. 두 경우 모두 전체 시스템의 total throughput은 12 Mbps로 동일하며, 출력포트 out_1은 10 Mbps를 서비스하고 있고, out_2는 2 Mbps를 서비스하고 있으므로 출력포트 사이의 공정성은 아무런 문제가 되지 않는다. 그러나 (a)의 경우 입력포트 in_1은 전체 4 Mbps를 in_2는 8 Mbps를 서비스하고 있으며, (b)의 경우 in_1은 전체 7 Mbps를 in_2는 5 Mbps를 서비스하고 있다. 직관적으로는 (b)의 경우가 입력포트들이 더 공정하게 서비스 받고 있음을 알 수 있다. 이러한 직관을 반영하기 위해서 정규 자원 할당 지수를 다음과 같이 정의할 수 있다.

$$x_i = \frac{\sum_j y_{i,j}}{\min\left(\sum_j z_{i\_in}, \sum_j z_{i,k\_out}\right)} \quad (6)$$

이때, 개체를 나타내는 x_i는 i번째 입력포트 혹은 출력포트를 나타내며, $y_{i,j}$는 i번째 입력포트에 존재하는 커넥션들, 즉 부 개체들을 나타내는 것이다. 그리고 $z_{i,j}$는 i번째 입력포트에서 j번째 부 개체에 할당된 서비스 능력을 나타낸다. 입력포트 및 출력포트 사이의 공정성 지수는 식 (6)을 식 (1)에 대입함으로써 구할 수 있다. 식 (6)에 있는 정의를 사용하여 그림 5에 제시된 입력포트들 사이의 공정성을 구하면 다음과 같다.

그림 5 (a)의 경우:

$$x_1 = \frac{2+2}{\min((8+2),(2+10))} = \frac{4}{10} = 0.4,$$

$$x_2 = \frac{8}{\min(10, 8)} = \frac{8}{8} = 1.0,$$

$$\text{fairness index} = \frac{(1.4)^2}{2\times(0.16+1.0)} = 0.845$$

그림 5 (b)의 경우:

$$x_1 = \frac{5+2}{\min((8+2),(5+10))} = \frac{7}{10} = 0.7,$$

$$x_2 = \frac{5}{\min(10,5)} = \frac{5}{5} = 1.0,$$

$$\text{fairness index} = \frac{(1.7)^2}{2\times(0.49+1.0)} = 0.97$$

위의 계산결과에서 보이는 것처럼 그림 5 (b)에 보이는 자원의 할당이 더 공정한 것이며, 이는 우리의 직관과 일치함을 나타내고 있다.

이러한 개념은 더욱 복잡하게 구성된 시스템에도 적용할 수 있다. 식 (6)의 분모를 식 (5)에 나와 있는 것처럼 복잡한 시스템을 구성하는 각 서비스 개체들 전체의 서비스 능력 중에서 가장 작은 것을 선택하면 된다. 그러나 이런 시스템은 현실성이 없는 것들이므로 이에 관한 논의는 더 이상 하지 않기로 한다.

VI. 결 론

현재까지의 공정성에 대한 연구는 시스템의 효율성과 같은 다른 성능 특성들과는 구분되어 독립적으로 연구되어 왔다. 이런 점을 감안하면 공정성과 효율성을 동시에 고려하는 것은 다소 이론의 소지가 있을 수도 있다. 그러나 다른 성능 특성들을 무시한 채 공정성만을 고려하는 것은 전체 시스템을 비효율적으로 만드는 등 실제적인 활용 범위에 있어 제한되게 된다. 대표적인 예가, 기존의 공정성 개념을 스위치에 적용할 때 나타난다. 출력 큐잉 스위치에서처럼 독립적인 부 시스템에 대해서는 공정성을 고려하는 것이 전체 시스템의 효율성(utilization)과 같은 다른 성능 지표에 영향을 주지 않지만, 입력 큐잉 스위치에서처럼 부 시스템들이 상호 의존적인 경우에는 전체 시스템의 효율성을 심각하게 저하시킬 수도 있다.

따라서 본 논문에서는 상관관계가 있는 즉, 독립적이지 않는 시스템에서 공정성을 평가하는 방법에 대해 고찰하였다. 입력 큐잉 스위치에서처럼 상관관계가 있는 시스템에서 특정한 부 시스템의 공정성을 평가할 때는 Jain이 정의한 공정성 및 공정성 지수를 그대로 사용할 수 없는 경우가 발생한다. 따라서 보다 포괄적으로 사용할 수 있는 공정성 및 공정성 지수를 정의할 필요가 있다. 본 논문에서는 정규 자원 할당 변수 x_i를 새로이 정의함으로써 기존의 Jain이 정의한 공정성 지수의 형태 및 공정성 지수가 가져야하는 다섯 가지 특성을 그대로 유지하면서 시스템 자원을 최대로 사용할 수 있도록 했다. 이는 공정성 지수가 가져야하는 여섯번째 특성으로“ 시스템의 효율성의 최대화 ”가 추가되어야 함을 의미한다.

이러한 개념은 네트워크와 같이 복잡한 공유 시스템을 지나가는 커넥션들의 공정성을 이야기할 때도 확장해서 사용할 수 있다. 이러한 경우에는 여러 노드에서 할당하는 자원 중 최소값에 대해 실제로 서비스 되는 비율을 정규 자원 할당 값으로 사용하면 된다. 본 논문에서 유도된 공정성의 정의는 공유 시스템의 자원을 공정하면서도 효율적으로 할당할 수 있는 기준으로 사용될 수 있다.

[참고문헌]

[1] R. Jain, D. Chiu, and W. Hawe, "A quantitative measure of fairness and discrimination for resource allocation in shared computer system," *Digital Equipment Corporation, Technical Report*, DEC-TR-301, Sep. 26, 1984. Available at http://www.cis.ohio-state.edu/~jain/papers/fairness.htm

[2] ATM Forum Document Number: ATM Form/94-0881. Available at http://www.cis.ohio-state.edu/~jain/

[3] ATM Forum Document Number: ATM Forum/99-0045. Available at http://www.cis.ohio-state.edu/~jain/

[4] M. Nabeshima and N. Yamanaka, "New scheduling mechanisms for achieving fairness criteria (MCR plus equal share, maximum of MCR or Max-Min share)," *IEICE Tr. Commun.*, Vol. E82-B, No. 5, June 1999, pp. 962-966.

[5] ATM Forum, "Traffic management specification 4.0," af-tm-0056.0, 1996.

[6] Hakyong Kim, Y.T. Lee, and K.S. Kim, "Fairness concept in terms of utilization," *IEE Electronics Letters*, Vol. 36, No. 4, Feb. 17, 2000, pp. 379-381.

[7] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*, 3rd ed., McGraw-Hill, Inc., 1991.

[8] Shang-Tse Chuang, A Goel, N. McKeown, and B. Prabhakar, "Matching output queueing with a combined input/ output-queued switch," *IEEE JSAC*, Vol. 16, No. 6, June 1999, pp. 1030-1039.

Fairness Evaluation in the Non-Independent Systems

Fairness is one of the important considerations in the performance study of shared systems. To date, fairness of a

shared system has been evaluated quantitatively by using the well-known fairness index defined by Jain in 1984. However, the fairness index defined by Jain can be applied limitedly to the fairness evaluation of a single and independent shared system. When we try to apply Jain's fairness index to a non-independent shared system where sub-systems have inter-dependency, it causes a problem of underutilization of the system resources. In this paper, therefore, we newly define a fairness index based on the Jain's approach so as to evaluate the fairness of the non-independent shared system such as input-queueing switch. The fairness index defined in this paper preserves all the characteristics of Jain's fairness definition and makes it possible to efficiently use the system resource.

Keywords : fairness, fairness index

김 학 용
(Hakyong Kim)

1995: 충남대학교 전자공학과 학사
1997: 광주과학기술원 정보통신공학과 석사
2001: 광주과학기술원 정보통신공학과 박사
2000: 일본 통신종합연구소(CRL) 방문연구원
2001. 9 2001. 12: 광주과학기술원 초고속 광네트워크 연구센터(UFON) 연구전담교수
2000 2001: IEEE Kwang-Ju Section, Chair of Student Activities
2000 현재: IEEE Tr. Communications, IEEE Communications Letters 논문심사 (Reviewer)
2000 2002: IEEE ICATM '01, IEEE ICON '01 및 ECUMN '02 TPC 멤버
2001. 12 현재: 코어세스 연구소 선임연구원
관심분야: IP/Ethernet QoS, DiffServ, Resilient Packet Ring, 고속 패킷 스위치 구조, 다중 입력 큐잉 방식 및 스케쥴링 알고리즘, 광 패킷 스위치 구조, IP over WDM 네트웍
E-mail: hykim@ieee.org
Tel: +82-2-3016-6702
Fax: +82-2-3016-6677

김 기 선
(Kiseon Kim)

1978: 서울대학교 전자공학과 학사
1980: 서울대학교 전자공학과 석사
1987: University of Southern California, Electrical Engineering 박사
1987 1989: Communications Sciences Institute, Research Scientists
1989 1991: Schlumberger Inc. Senior Development Engineer
1991 1994: Superconducting Super Collider Lab., Computer Communication Specialist
1994 현재: 광주과학기술원 정보통신공학과 교수
관심분야: 디지털 통신 이론 및 응용, 멀티미디어 통신 신호 처리 및 VLSI 구현, 원격 측정 및 장비, 광대역 가입자용 디지털 모뎀
E-mail: kskim@kjist.ac.kr
Tel: +82-62-970-2207
Fax: +82-62-970-2204