

# 멀티홉 무선 네트워크에서 큐 결합에 의한 패킷 전송 지연의 영향

종신회원 최진구\*

## Effect of Queue Aggregation on Packet Transmission Delay in Multi-hop Wireless Networks

Jin-Ghoo Choi\* *Lifelong Member*

요약

패킷 스케줄링 기법에 대한 연구는 백프레셔 알고리즘으로부터 시작하여 최근에는 CSMA/CA 기반의 분산 최대수율 스케줄러에까지 이르렀다. 그러나, 지금까지 알려진 스케줄링 기법들은 각 링크에서 세션마다 독립된 큐를 유지하며 이로 인해 네트워크의 확장성이 제한되고 있다. 가상 큐 방식으로 동일 링크에 위치한 세션 큐를 결합하는 아이디어가 제안되었으나 이에 따른 패킷 지연의 영향은 아직 알려지지 않고 있다. 본 논문에서는 간단한 정상 랜덤 스케줄러에 대한 분석을 통해 큐 결합과 패킷 지연의 관계를 정량적으로 살펴본다. 먼저, 기존의 세션 기반 랜덤 스케줄러를 분석하여 정확한 평균 패킷 지연을 구하고, 같은 링크 상의 세션 큐들을 결합하여 링크 기반의 랜덤 스케줄링으로 확장한다. 두 스케줄링 방식의 패킷 지연을 비교함으로써 큐 결합에 의해 네트워크의 패킷 전송 시간이 감소할 수 있음을 보인다.

**Key Words** : Stationary Randomized Scheduling (SRS), Queue aggregation, Throughput optimality, Average packet delay, Multi-hop wireless network

### ABSTRACT

Since the introduction of the back-pressure scheduling policy, tremendous efforts have been devoted to the packet schedulers achieving the optimal throughput in multi-hop wireless networks. Our understanding for the throughput performance has been matured now and the focus is migrating to the delay. Unfortunately, we have witnessed very limited success on the delay analysis of multi-hop networks even in the wireline domain. Noting that the previous literatures on schedulers commonly assume the separate session queues at each link, we are interested in the effect of queue aggregation on the delay performance of networks. For this purpose, we first analyze the stationary randomized scheduling policy with session queues (SRS-S). Then, the scheduler is modified to manage only a single queue at links, which is called the stationary randomized scheduling with link queues (SRS-L). By comparing the delay performances of SRS-S and SRS-L, we show that the queue aggregation reduces the average delay of packets significantly.

### 1. 서론

멀티홉 무선 네트워크는 셀룰러 시스템과 같은 단

일홉 네트워크에 비해 잠재적으로 높은 용량을 가지고 있어 학계와 산업계로부터 많은 관심을 받아 왔다<sup>1)</sup>. 그러나, 멀티홉 네트워크의 잠재적인 용량에도 불구하

※ 이 논문은 2008년 정부(교육과학기술부)의 재원으로 한국학술진흥재단의 지원을 받아 수행된 연구임 (KRF-2008-357-D00171).

\* 영남대학교 정보통신공학과(jchoi@yu.ac.kr)

논문번호 : KICS2010-01-016, 접수일자 : 2010년 1월 12일, 최종논문접수일자 : 2010년 9월 15일

고 날로 증가하는 데이터 전송 속도에 대한 수요를 감당하기에는 충분하지 않으며, 무선 자원을 효율적으로 활용하는 문제는 여전히 중요하다. 네트워크에 대한 효율 이론에 따르면 MAC 계층의 패킷 스케줄러는 전송 계층의 혼잡 제어기(congestion controller)와 협력하여 네트워크의 효율을 극대화할 수 있다<sup>2)</sup>. 패킷 스케줄러는 네트워크 시스템에서 가장 많은 자원을 소모하는 구성 요소이므로 효율적인 스케줄링 기법은 네트워크 성능 향상에 있어 필수적이다

백프레서(back-pressure) 스케줄러가 알려진 이후 최대수율(throughput-optimal) 스케줄링 기법에 대해 많은 연구가 집중되어 왔다<sup>3)~7)</sup>. 백프레서 스케줄러는 네트워크의 수율을 한계까지 높일 수 있으며 입력 트래픽에 동적으로 적응하는 장점이 있으나, 매 시간 슬롯마다 NP-hard로 알려진 MWM (Maximum Weighted Matching) 문제를 풀어야 하는 부담이 있고 중앙 집중적인 제어가 필요하며 패킷 지연 성능을 알기 어려운 단점이 있다. 이에 대한 해결 방법으로 MWM을 MM (Maximal Matching)이나 GMM (Greedy Maximal Matching)으로 근사하거나<sup>4)</sup> 랜덤 알고리즘에 의한 점진적으로 최적화하는 기법이 제안되었으며<sup>5)</sup>, CSMA/CA 방식의 분산 제어를 통해 백프레서와 유사하게 동작하는 알고리즘도 개발되었다<sup>6)</sup>.

반면 멀티홉 네트워크에서 패킷 스케줄링 기법의 지연 성능에 대한 분석은 아직 만족스러운 결과가 알려지지 않고 있다. 현재까지의 연구 중에서 가장 주목할 만한 성과는 Lyapunov 함수의 변화율로부터 패킷 지연의 상계를 구하는 방법이며<sup>7)</sup>, 이에 의해 정상 랜덤 스케줄링(stationary randomized scheduling: SRS)의 평균 패킷 지연이 분석되었다. 임의의 스케줄러에 대한 패킷 지연의 하계를 얻을 수 있는 방법도 발표되었으나, 경우에 따라 무한한 오차를 가질 수 있는 단점이 있다<sup>8)</sup>. 그 밖에 Large deviation 이론을 통한 분석이 시도되고 있으나 복잡도가 높아 멀티홉 네트워크에서는 적용하기 어려울 것으로 보인다.

본 논문의 주요 관심은 큐의 결합이 패킷 지연에 미치는 영향을 살펴보는 것이다. 백프레서를 포함하여 지금까지 알려진 최대수율 스케줄러들은 각 링크마다 독립된 세션 큐가 있다고 가정하였다. 이에 따라 서로 다른 세션 사이의 상호 작용을 무시할 수 있어 스케줄링의 성능 분석이 용이하였다. 그러나, 실용적인 측면에서는 세션 큐의 유지에 필요한 연산 부담이 증가하여 네트워크 확장성이 저하되는 단점이 있다. 이론적인 측면에서 더욱 중요한 문제는 세션 큐를 유지함으로써 스케줄러의 성능, 특히 패킷 지연 성능에 어떤

영향을 미치는지 명확하지 않다는 것이다.

본 연구에서는 정상 랜덤 스케줄러에 대한 분석을 통해 이에 대한 부분적인 답을 얻고자 한다. 널리 알려진 랜덤 스케줄링 기법을 세션 기반의 정상 랜덤 스케줄러(SRS with Session queues: SRS-S)라고 정의하고, Lyapunov 변화율을 응용한 새로운 방법으로 패킷 지연 성능을 분석한다. 다음으로, 동일 링크에 위치한 모든 세션 큐를 결합하여 링크마다 단일 큐를 유지하는 랜덤 스케줄링 기법을 제안하고 링크 기반의 정상 랜덤 스케줄러(SRS with Link queues: SRS-L)라고 정의한다. 이 스케줄러의 평균 패킷 지연 역시 동일한 방법으로 분석할 수 있으며, 두 랜덤 스케줄러의 성능을 비교함으로써 큐 결합과 패킷 지연의 관계를 정량적으로 보인다. 간단한 네트워크 토폴로지에서도 SRS-L은 SRS-S보다 25~40% 패킷 지연이 작음을 확인할 수 있다.

이하 2절에서 시스템 모델과 네트워크의 안정성, 용량의 개념을 정의한다. 3절에서는 SRS-S 스케줄링을 정의하고 분석하며 기존의 결과와 비교한다. 4절에서는 SRS-L 스케줄링을 정의하고 분석하며, SRS-S와 패킷 지연 성능을 비교한다. 5절에서는 간단한 예를 통해 SRS-L 기법의 지연 성능이 SRS-S보다 우월함을 보이며, 6절에서 결론을 맺는다.

## II. 모델 및 정의

### 2.1 시스템

본 논문에서 고려하는 멀티홉 무선 네트워크는  $N$ 개의 노드와  $L$ 개의 방향성 링크를 가진 그래프로 표현된다. 링크는 동일한 간격의 시간 슬롯으로 구성되어 있으며, 모든 링크에서 슬롯의 시작 시점은 동기화되어 있다고 가정한다. 임의의 슬롯에서 패킷 스케줄러에 의해 선택된 링크는 큐에서 대기하고 있는 패킷 중에서 한 개를 전송하며, 큐가 비어 있으면 아무 것도 전송하지 않는다. 이 때, 전송된 패킷은 오류 없이 수신단에 전달된다고 가정하자. 일반적으로 무선 환경에서는 무시할 수 없는 확률로 전송 오류가 발생하지만, 본 논문의 분석은 랜덤 오류가 있는 경우에도 쉽게 확장될 수 있으므로 설명의 편의를 위해 고려하지 않는다.

유선 네트워크와 달리 무선 네트워크에서는 활성화된 링크 사이에 간섭이 발생하므로 모든 링크가 동시에 서비스될 수는 없다. 본 논문에서 링크의 활성화와 서비스는 동일한 의미로 사용된다. 무선 링크 사이의 간섭에 대해 다양한 모델들이 제안되어 왔으나, 멀티

홉 무선 네트워크에서 스케줄링에 관한 기존 연구들은 주로  $K$ 홉 간섭 모델을 채택해 왔다.  $K$ 가 1인 경우는 노드 배타적(node exclusive) 모델이라고도 불리며, 송신 노드 혹은 수신 노드가 동일한 링크들은 동시에 서비스될 수 없다고 가정한다. 이러한 모델은 블루투스(Bluetooth)나 FH-CDMA 네트워크를 기술하는데 있어 적합한 것으로 알려져 있다<sup>6)</sup>. 또한,  $K$ 가 2인 경우의 간섭 모델은 RTS/CTS 메커니즘을 적용한 IEEE802.11 DCF 네트워크를 기술하는데 있어 적합하다. 본 논문에서는 동시에 서비스될 수 있는 링크의 집합을 모두 명시하며 이것은  $K$ 홉 모델을 포괄하는 일반적인 간섭 모델이다. 링크 서비스 벡터  $\vec{\pi} := (\pi_1, \dots, \pi_L)$ 는 임의의 슬롯에서 각 링크의 활성화 여부를 나타내며,  $\pi_l = 1$ 은 링크  $l$ 이 활성화됨을,  $\pi_l = 0$ 은 활성화되지 않음을 의미한다. 네트워크에서 허용되는 모든 링크 서비스 벡터의 집합을  $\mathbf{F}$ 로 정의하자.

외부로부터 임의의 출발 노드에 인가된 패킷들이 네트워크를 경유하여 미리 정의된 도착 노드로 이동하는 일련의 흐름을 세션이라고 정의하고, 네트워크 상의 모든 세션 수를  $S$ 로 나타낸다. 슬롯  $n$ 에서 출발 노드에 입력된 세션  $s$ 의 패킷 수를  $A^s[n]$ 이라고 하자. 이 때, 1)  $A^s[1], A^s[2], \dots$ 는 정상 랜덤 프로세스를 형성하고, 2) 각 슬롯에서의 패킷 수는 서로 *i.i.d.*이며, 3)  $EA^s[n]$ 와  $E(A^s[n])^2$ 는 유한하다고 가정한다. 또한, 세션  $s$ 의 패킷 도착률을  $\lambda^s := EA^s[n]$ 으로 나타내자. 패킷은 세션마다 정의된 라우팅 경로를 따라 한 링크씩 이동하며, 먼저 링크에 위치한 큐에서 대기하다가 스케줄러에 의해 그 큐가 서비스되는 순간 전송된다. 따라서, 패킷은 하나의 링크를 건널 때마다 최소 한 슬롯의 지연을 겪는다.

세션  $s$ 의 라우팅 경로는  $H^s$ 개의 연속된 링크로 구성되며, 경로 상의  $i$ 번째 링크를  $(s, i)$ 로 표기하자. 슬롯  $n$ 에서 링크  $(s, i)$ 에 대기 중인 세션  $s$ 의 패킷 수를  $Q_i^s[n]$ 이라고 하자. 여러 세션이 동일한 링크를 경유할 수도 있으므로 하나의 링크  $l$ 에 대응하는  $(s, i)$ 는 복수일 수 있으며, 이러한  $(s, i)$ 를 모아 집합  $\mathcal{S}_l$ 라고 정의한다. 어떤 세션도 경유하지 않는 링크는 고려할 필요가 없으므로  $\mathcal{S}_l$ 의 원소 개수는 항상 1보다 크거나 같다고 가정한다. 이 때, 링크  $l$ 에서 대기 중인 모든 패킷의 수는  $Q_l[n] = \sum_{(s,i) \in \mathcal{S}_l} Q_i^s[n]$ 으로 주어짐을 알 수 있다. 네트워크 내의 모든 큐의 길이

를 벡터  $\vec{Q}[n]$ 으로 나타내며, 슬롯 인덱스가 없는  $Q_i^s$ 와  $Q_l$ 은 큐의 길이가 아닌 큐 자체를 가리킨다.

슬롯  $n$ 에서 스케줄러에 의해  $Q_i^s$ 가 서비스되면  $\pi_i^s[n] = 1$ , 그렇지 않으면  $\pi_i^s[n] = 0$ 으로 나타낸다. 또한,  $Q_i^s$ 가 선택되었을 때 대기 중인 패킷이 있어 실제 전송이 발생하면  $\mu_i^s[n] = 1$ , 그렇지 않으면  $\mu_i^s[n] = 0$ 으로 나타낸다. 표기의 편의상  $\pi_0^s[n] = \mu_0^s[n] := A^s[n]$ 로 정의하자. 동일한 방식으로  $Q_l$ 의 서비스 여부를 나타내는  $\pi_l[n]$ 과 실제 전송 여부를 나타내는  $\mu_l[n]$ 을 정의한다.

### 2.2 네트워크 안정성

**정의1:** 슬롯  $n$ 에서 네트워크 내의 모든 큐 길이의 합을  $Q_{tot}[n]$ 이라고 할 때,

$$\limsup_{m \rightarrow \infty} \frac{1}{m} \sum_{n=1}^m EQ_{tot}[n] < \infty$$

이면 그 네트워크는 안정적 (stable)이라고 정의한다.

위의 정의에 따르면 네트워크가 안정적이기 위해서는 모든 큐가 동시에 안정되어야 한다. 시간에 따라 무한히 커지는 큐가 하나라도 존재하면  $EQ_{tot}[n]$  역시 무한히 증가하고, 그 시평균도 무한하기 때문이다. 또한,  $EQ_{tot}[n]$ 의 시평균이 수렴하지 않고 진동하더라도 무한으로 발산하지만 않으면 시스템은 안정적이라고 간주한다. 즉, 임의의 패킷이 유한한 시간 내에 도착 노드에 도달함을 보장하면 그 네트워크는 안정적이다.

### 2.3 네트워크 용량

벡터  $\vec{\lambda} := (\lambda^1, \dots, \lambda^S)$ 는 모든 세션들의 패킷 도착률을 나타낸다. 주어진 네트워크에 대해  $\vec{\lambda}$ 가 일정한 한계를 넘어서면 시스템이 불안정해 짐을 쉽게 예상할 수 있다. 라우팅 경로와 링크 전송속도가 고정되어 있을 때, 패킷 도착률  $\vec{\lambda}$ 의 트래픽을 안정적으로 처리할 수 있는 스케줄러가 하나라도 존재하면 그 트래픽은 네트워크에 의해 수용 가능하다고 표현하며, 이러한  $\vec{\lambda}$ 의 집합을 네트워크 용량 (capacity region)이라고 정의한다.

네트워크 용량  $\Lambda$ 를 수학적으로 정의하기에 앞서 링크 서비스율 영역  $\Pi$ 를 먼저 정의하자.  $\mathbf{F}$ 는 동시에

활성화될 수 있는 링크들을 가리키는 벡터  $\vec{\pi}^{(k)}$ 의 집합이라고 정의하였다. 링크 서비스율 영역  $\Pi$ 는  $F$ 의 컨벡스 헐 (convex hull)

$$\{ \sum_k p^{(k)} \vec{\pi}^{(k)} \mid 0 \leq p^{(k)} \leq 1, \sum_k p^{(k)} = 1, \vec{\pi}^{(k)} \in F \}$$

로 정의되며,  $p^{(k)}$ 는 벡터  $\vec{\pi}^{(k)}$ 가 선택될 확률을 의미한다. 이 때, 네트워크 용량  $\Lambda$ 는 다음과 같이 정의된다.

$$\{ \vec{\lambda} \mid \vec{\lambda}' := (\lambda_1', \dots, \lambda_L') \in \Pi, \lambda_l' = \sum_{s \mid (s,i) \in \mathcal{S}_l} \lambda_s^s, \forall l \}$$

위의 정의에서 벡터  $\vec{\lambda}'$ 의  $l$ 번째 원소  $\lambda_l'$ 은 링크  $l$ 에서의 패킷 도착률을 가리킨다. 단, 네트워크가 안정적으로 동작하여, 각 링크에 도달한 패킷이 유한한 시간 내에 떠난다고 가정할 때의 도착률이므로 명목상 패킷 도착률이라고 정의하자.

패킷 도착률이 네트워크 용량에 포함되지 않는 경우, 허용 가능한 링크 서비스 벡터들을 어떻게 조합하더라도 그 트래픽을 수용할 수 없으므로 어떤 스케줄링 기법도 네트워크를 안정화시킬 수 없다. 즉,  $\vec{\lambda} \in \Lambda$ 는 네트워크가 안정화되기 위한 필요조건이다. 반면  $\vec{\lambda} \in \Lambda$ 가 네트워크 안정화를 위한 충분조건인지는 직관적으로 알기 어렵다. 그러나, Tassiulas는 백프레서라고 알려진 스케줄링 기법을 고안함으로써 단일 스케줄러에 의해 네트워크 용량 내의 임의의 트래픽을 수용할 수 있음을 증명하였다<sup>31</sup>. 이러한 스케줄러들은 최대수율을 갖는다고 정의되며, 최대수율 스케줄링 기법을 적용한다는 가정하에서  $\vec{\lambda} \in \Lambda$ 는 네트워크 안정화를 위한 충분조건이다.

### III. 세션 기반의 정상 랜덤 스케줄링

정상 랜덤 스케줄링이란 미리 정해진 확률에 따라 매 슬롯마다 링크 서비스 벡터를 선택하고 그 벡터가 가리키는 링크들을 활성화시키는 스케줄링 방식이다. 특히, 각 링크에서 세션마다 독립된 큐를 유지하는 경우 세션 기반의 정상 랜덤 스케줄링 (SRS-S)이라고 정의하자. SRS-S 스케줄러는 서비스할 링크의 선택이 큐의 길이와 무관하며, 세션마다 분리된 큐를 가지고

있어 세션 간 상호 작용이 없으므로 분석이 비교적 용이하다. 기존에 분석된 SRS-S의 패킷 지연은 험거운 상계일 뿐이므로<sup>17,11</sup>, 큐 결합과 패킷 지연의 관계를 살펴보는 데에는 충분하지 않다. 이번 장에서는 SRS-S 스케줄링을 정의하고 그 평균 패킷 지연을 정확히 계산하는 분석 방법을 제시한다.

#### 3.1 스케줄링 기법 및 DTMC 모델

SRS-S 스케줄링을 적용하려면 각 세션은 라우팅 경로 상의 모든 링크에서 다른 세션과 분리된 큐를 가져야 하며, 패킷 도착률  $\vec{\lambda}$ 를 미리 알아야 한다.  $\vec{\lambda}$ 를 미리 알 수 없는 경우 스케줄링을 수행하는 동시에 동적으로 트래픽을 측정하여 스케줄링 파라미터를 적용할 수도 있으나 본 논문의 범위를 벗어나므로 상세한 설명은 생략한다.  $\vec{\lambda} \in \Lambda$ 라고 가정하고, 이에 대응되는 각 링크에서의 명목상 패킷 도착률을  $\vec{\lambda}'$ 라고 하자. SRS-S 스케줄링은 오프라인으로 수행되는 준비 단계와 매 슬롯 반복적으로 수행되는 동작 단계로 구분할 수 있으며, 각 단계의 절차는 다음과 같다.

##### 준비 단계

1) 초과 서비스율의 선택: SRS-S 스케줄러는  $\vec{\lambda}' + \epsilon_{SRS-S}(|\mathcal{S}_1|, \dots, |\mathcal{S}_L|) \in \Pi$ 를 만족하는 임의의 양의 실수  $\epsilon_{SRS-S}$ 를 선택한다.  $|\mathcal{S}_l|$ 는  $\mathcal{S}_l$ 의 원소 개수이므로 결국 링크  $l$ 을 경유하는 세션의 수를 의미한다.  $\epsilon_{SRS-S}$ 는 SRS-S 스케줄러가 각 세션 큐에게 제공하는 초과 서비스율에 해당하며,  $\vec{\lambda}$ 가  $\Lambda$ 의 경계에 있지 않다면 반드시 존재한다.

2) 링크 서비스 벡터 선택 확률의 계산:  $\Pi$ 의 정의에 따르면

$\vec{\lambda}' + \epsilon_{SRS-S}(|\mathcal{S}_1|, \dots, |\mathcal{S}_L|) = \sum_k \vec{\pi}^{(k)} p_{SRS-S}^{(k)}$ 를 만족하는  $p_{SRS-S}^{(k)}$ 들이 존재하며 스케줄러는 이 확률을 미리 계산한다.

##### 동작 단계

1) 활성화 링크 선택: 준비 단계에서 계산한  $p_{SRS-S}^{(k)}$ 의 확률에 따라 링크 서비스 벡터  $\vec{\pi}^{(k)}$ 를 선택하고,  $\pi_l^{(k)} = 1$ 인 링크  $l$ 들을 활성화한다. 이 때, 링크  $l$ 의 활성화 확률은  $\lambda_l' + \epsilon_l = \sum_{s \mid (s,i) \in \mathcal{S}_l} (\lambda_s^s + \epsilon_{SRS-S})$ 임을 알 수 있다.

2) 활성화 세션 선택: 활성화된 링크  $l$ 은  $\lambda_s^s + \epsilon_{SRS-S}$ 의 확률로 자신을 경유하는 세션  $s$ 를 랜

덤하게 선택한다. 이에 따라 링크  $l$ 에 있는 세션  $s$ 의 큐는  $\lambda^s + \epsilon_{SRS-S}$ 의 서비스율을 얻는다.

SRS-S 스케줄링에서  $\vec{\lambda} \in A$ 이면 네트워크가 안정적임을 다음의 설명으로부터 알 수 있다. 임의의 세션  $s$ 의 첫 번째 큐는 네트워크 외부로부터  $\lambda^s$ 의 트래픽을 입력 받으며, 스케줄러는  $\lambda^s + \epsilon_{SRS-S}$ 의 속도로 그 큐를 서비스한다. 패킷 도착률보다 서비스율이 크므로 모든 패킷은 유한한 시간 내에 첫 번째 큐를 떠나  $\lambda^s$ 의 속도로 두 번째 큐에 인가된다. 두 번째 큐의 서비스율 역시  $\lambda^s + \epsilon_{SRS-S}$ 이므로 모든 패킷은 유한한 시간 내에 큐를 떠나  $\lambda^s$ 의 속도로 세 번째 큐를 향한다. 이러한 과정을 반복함으로써 세션  $s$ 의 모든 큐가 안정적임을 알 수 있으며, 임의의 세션에 속하는 모든 큐가 안정적이므로 네트워크도 안정적이다.

다음으로 SRS-S 스케줄링에 대한 DTMC (Discrete Time Markov Chain) 모델을 수립하자. 다음 절에서 설명할 패킷 지연 분석 방법은 이 DTMC의 Lyapunov 변화율을 이용한다. 슬롯  $n$ 에서 네트워크 내의 모든 큐 길이를 벡터  $Q_{SRS-S}^s[n] := (Q_i^s[n])$ 으로 나타내자. 이 때, 세션 큐  $Q_i^s$ 의 길이는 시간에 따라  $Q_i^s[n+1] = Q_i^s[n] + \mu_{i-1}^s[n] - \mu_i^s[n]$ 로 변화한다. 다음 슬롯에서의 큐 길이  $Q_{SRS-S}^s[n+1]$ 는 현재 슬롯의 정보인  $Q_{SRS-S}^s[n]$ ,  $A^s[n]$ ,  $\mu_i^s[n]$ 에 의해서 결정되므로  $Q_{SRS-S}^s[n]$ 은 DTMC임을 알 수 있다.

다음의 명제는 Foster 정리<sup>[9]</sup>에 의해 증명할 수 있으며, DTMC가 양의 회귀성(positive recurrence)을 가지면 정 1의 안정성도 당연히 성립하므로, 명제 1은 SRS-S 스케줄링의 안정성을 내포한다.

**명제1:** SRS-S 스케줄링에 대한 DTMC  $Q_{SRS-S}^s[n]$ 은 양의 회귀성을 가지며, 따라서 정상 확률분포가 존재한다.

### 3.2 패킷 지연

이번 절에서는 SRS-S 스케줄러의 평균 패킷 지연을 분석하는 새로운 방법을 제시하고 이에 따른 결과를 기존의 상계와 비교한다. 이 기법은 정상 상태의 DTMC에 대해서만 적용 가능한 한계가 있으나 기존의 분석보다 간결하면서도 정확한 결과를 얻을 수 있으며, 명제1에 의해 SRS-S에도 적용할 수 있다.

SRS-S 스케줄러의 평균 패킷 지연에 대한 분석은 다음의 명제로부터 출발한다.

**명제2:** DTMC  $Q[n]$ 이 양의 회귀성을 가지면,

Lyapunov 함수  $V(\cdot)$ 의 변화율은 정상 상태에서 0이다.

**증명:**  $Q[n]$ 이 양의 회귀성을 가지면 정상 확률분포가 존재하므로

$$\begin{aligned} EV(Q[n+1]) &= \sum_q V(q) \Pr Q[n+1] = q \\ &= \sum_q V(q) \Pr Q[n] = q \\ &= EV(Q[n]) \end{aligned}$$

이 성립한다. 따라서,

$$E(V(Q[n+1]) - V(Q[n])) \equiv 0 \text{이다.} \quad \blacksquare$$

랜덤 스케줄링에서 활성화될 링크 혹은 큐는 미리 정해진 확률에만 의존하며 큐의 길이와는 무관하다. 이러한 특성은 랜덤 스케줄링에 대한 분석을 용이하게 한다. 아래의 명제는 SRS-S의 DTMC  $Q_{SRS-S}^s[n]$ 에 대한 Lyapunov 변화율을 계산하는 데에 사용된다.

**명제3:**  $\vec{\lambda} \in A$ 일 때, SRS-S 스케줄링에 대해 다음의 두 식이 성립한다.

- 1)  $E(Q_i^s[n] \mu_{i-1}^s[n]) = EQ_i^s[n] \lambda^s$ ,
- 2)  $E(Q_i^s[n] \mu_i^s[n]) = EQ_i^s[n] (\lambda^s + \epsilon_{SRS-S})$ .

**증명:** 1)  $Q_{i-1}^s$ 의 서비스 여부는  $Q_i^s$  길이와 무관하므로  $E(Q_i^s[n] \mu_{i-1}^s[n]) = EQ_i^s[n] E\mu_{i-1}^s[n]$ 이며, SRS-S를 적용한 네트워크는  $\lambda \in A$ 에 대해 안정적이므로  $E\mu_{i-1}^s[n] = \lambda^s$ 이다. 따라서, 1)이 성립한다.

2)  $Q_i^s$ 의 서비스 여부는 큐 길이와 무관하나 스케줄러에 의해 선택되었을 때 대기 중인 패킷이 없으면 전송이 일어나지 않는다. 따라서,  $\mu_i^s[n]$ 는  $Q_i^s[n]$ 와 약한 상관관계를 갖는다.  $Q_i^s[n] = 0$ 일 때와  $Q_i^s[n] > 0$ 일 때로 나누어 생각하면  $Q_i^s[n] \mu_i^s[n] = Q_i^s[n] \pi_i^s[n]$ 임을 확인할 수 있으며, 따라서  $E(Q_i^s[n] \mu_i^s[n]) = E(Q_i^s[n] \pi_i^s[n])$ 이다. 또한,  $\pi_i^s[n]$ 는  $Q_i^s[n]$ 와 무관하게 확률  $\lambda^s + \epsilon_{SRS-S}$ 로 선택되므로 2)가 성립한다.

다음의 정리는 SRS-S의 평균 패킷 지연을 제시하며 명제 2와 3을 이용하여 증명할 수 있다. 패킷 지연을 간단한 수식으로 나타내기 위해 각 노드의 라디오 인터페이스는 1개이고 따라서 동일 노드에 연결된 링크 중에서 하나만이 동시에 서비스될 수 있다고 가정하자. 이러한 가정 없이도 동일한 방법으로 패킷 지연을 얻을 수 있지만 대신 수식의 형태가 복잡하다.  $\lambda_{tot}$ 는 네트워크에 인가되는 전체 패킷의 도착률로서

$\sum_s \lambda^s$ 로 주어지며,  $\sigma^2(A^s)$ 는  $A^s[n]$ 의 분산을 뜻한다.

**정리1:**  $\vec{\lambda} \in \Lambda$ 일 때, SRS-S 스케줄러의 평균 패킷 지연은

$$\frac{\sum_s (\sigma^2(A^s) + (2H^s - 1)\lambda^s - (\lambda^s)^2)}{2\epsilon_{SRS-S}\lambda_{tot}}$$

이며,  $\hat{\epsilon}_{SRS-S} := \sup \epsilon_{SRS-S}$ 에서 최소화된다.

**증명:** 일반적으로 Lyapunov 함수는 동적인 시스템의 안정성을 보이기 위해 널리 사용되며, 다차원으로 주어진 시스템 상태를 일차원으로 매핑함으로써 안정성 분석을 용이하게 한다<sup>[10]</sup>. 그러나, 본 정리의 증명에서는 이미 안정적임이 알려져 있는 시스템에 대한 Lyapunov 함수의 변화율로부터 해당 시스템의 평균적인 큐 길이를 구하는 접근 방식이 사용된다.

SRS-S 스케줄링에 대한 DTMC 모델  $Q_{SRS-S}^s[n]$ 의 Lyapunov 함수를

$V(Q_{SRS-S}^s[n]) := 2^{-1} \sum_{(s,i)} (Q_i^s[n])^2$ 로 정의하면,  $V(\cdot)$ 의 변화율은 다음과 같다.

$$\begin{aligned} & \mathbf{E} V(Q_{SRS-S}^s[n+1]) - V(Q_{SRS-S}^s[n]) \\ &= \sum_{\vee (s,i)} \mathbf{E} Q_i^s[n] (\mu_{i-1}^s[n] - \mu_i^s[n]) + c_{SRS-S} \\ &= -\epsilon_{SRS-S} \sum_{\vee (s,i)} \mathbf{E} Q_i^s[n] + c_{SRS-S} \end{aligned}$$

위 식에서 마지막 등식은 명제 3에 의해 성립한다.  $c_{SRS-S} := 2^{-1} \sum_{(s,i)} \mathbf{E} (\mu_{i-1}^s[n] - \mu_i^s[n])^2$ 로 정의되었으며, 1) 노드 당 인터페이스 수가 1인 경우  $i > 1$ 일 때  $\mu_{i-1}^s = \mu_i^s = 0$ 이고, 2)  $i > 0$ 일 때  $(\mu_i^s)^2 = \mu_i^s$ 이므로

$$\begin{aligned} & c_{SRS-S} \\ &= 2^{-1} \sum_s \mathbf{E} (\mu_0^s - \mu_1^s)^2 + \sum_{i=2}^H (\mu_{i-1}^s - \mu_i^s)^2 \\ &= 2^{-1} \sum_s \mathbf{E} (A^s)^2 - 2A^s \mu_1^s + \mu_1^s \\ & \quad + \sum_{i=2}^H (\mu_{i-1}^s + \mu_i^s) \\ &= 2^{-1} \sum_s (\sigma^2(A^s) + (2H^s - 1)\lambda^s - (\lambda^s)^2) \end{aligned}$$

임을 알 수 있다. 위의 식에서 시간을 가리키는 인덱스  $n$ 은 생략되었다.

명제 2에 의해  $V(\cdot)$ 의 변화율은 0이므로 네트워크 내의 모든 큐 길이의 기대 값은

$\mathbf{E} Q_{tot}[n] = \epsilon_{SRS-S}^{-1} c_{SRS-S}$ 이며, Little의 정리에 의해 평균 패킷 지연은  $(\lambda_{tot} \epsilon_{SRS-S})^{-1} c_{SRS-S}$ 로 주어진다. 패킷 지연은  $\epsilon_{SRS-S}$ 에 따라 단조 감소하므로 최소의 지연을 얻기 위해서는 최대한 큰  $\epsilon_{SRS-S}$ 을 선택해야 한다. ■

SRS-S의 평균 패킷 지연에 대한 상계는 [7]와 [11]에서 분석되었으며 그 결과는 다음과 같다.

$$H_M \cdot \frac{N + \sum_s (\sigma^2(A^s) + \lambda^s - (\lambda^s)^2)}{2\epsilon_{SRS-S}\lambda_{tot}}$$

위 식에서  $H_M := \max H^s$ 는 가장 긴 라우팅 경로의 링크 수를 의미한다. 이 식은 정리 1의 결과와 유사하나, 노드의 수  $N$ 과 최장 라우팅 경로의 길이  $H_M$ 에 의존하고 있는 점에서 다르다. 다음의 명제에 의해 노드 수  $N$ 에의 의존성을 제거하자.

**명제4:** 각 노드의 라디오 인터페이스의 수가 1이면  $N \geq 2 \sum_s \lambda^s H^s$ 이다.

**증명:** 라디오 인터페이스 수가 1이므로, 각 노드의 패킷 입력 속도와 출력 속도의 합은 1보다 작거나 같다. 네트워크의 모든 노드를 고려하면 각 입력과 출력은 두 번씩 계산되므로  $N \geq 2 \sum_s \lambda^s H^s$ 임을 알 수 있다. ■

명제 4로부터 위의 패킷 지연은

$$\begin{aligned} & H_M \cdot \frac{N + \sum_s (\sigma^2(A^s) + \lambda^s - (\lambda^s)^2)}{2\epsilon_{SRS-S}\lambda_{tot}} \\ & \geq H_M \cdot \frac{\sum_s (\sigma^2(A^s) + (2H^s - 1)\lambda^s - (\lambda^s)^2)}{2\epsilon_{SRS-S}\lambda_{tot}} \end{aligned}$$

이며, 정리 1에서 제시한 SRS-S 스케줄러의 평균 패킷 지연은 기존의 결과보다 최소  $H_M^{-1}$ 만큼 작다.

#### IV. 링크 기반의 정상 랜덤 스케줄링

각 링크에서 세션마다 독립된 큐를 관리하는 경우 HOL (head-of-line) 봉쇄와 같은 세션 간 상호작용이 발생하지 않아 성능 분석이 용이하지만, 큐 관리로 인해 연산 능력의 소모가 크고 시스템이 복잡해지는 단점이 있다. 이번 장에서는 링크마다 단일 큐를 유지하는 정상 랜덤 스케줄링 기법(SRS-L)을 제시하고 그

스케줄러에 의해 얻을 수 있는 평균 패킷 지연을 분석하며, 동일 링크에 위치한 세션 큐의 결합이 패킷 전송 지연에 미치는 영향을 살펴본다.

4.1 스케줄링 기법 및 DTMC 모델

SRS-L 스케줄링을 적용할 때 각 링크에는 하나의 큐만이 존재하므로 같은 링크를 점유하는 패킷은 모두 동일한 큐에서 대기한다. SRS-S와 마찬가지로 패킷 도착률은  $\lambda$ , 각 링크에서의 명목상 패킷 도착률은  $\vec{\lambda}$ 로 표기하며  $\vec{\lambda} \in A$ 라고 가정한다. SRS-L 스케줄링이 준비단계와 동작단계에서 수행하는 작업은 다음과 같다.

준비 단계

1) 초과 서비스율의 선택: SRS-L 스케줄러는  $\vec{\lambda}' + \epsilon_{SRS-L}(1, \dots, 1) \in \Pi$ 를 만족하는 임의의 양의 실수  $\epsilon_{SRS-L}$ 를 선택한다.

2) 링크 서비스 벡터 선택 확률의 계산:  $\Pi$ 의 정의에 따라  $\vec{\lambda}' + \epsilon_{SRS-L}(1, \dots, 1) = \sum_k \pi^{(k)} p_{SRS-L}^{(k)}$ 를 만족하는  $p_{SRS-L}^{(k)}$ 가 반드시 존재하므로 스케줄러는 이 확률을 미리 계산한다.

동작 단계

1) 활성화 링크 선택: 준비 단계에서 계산한 확률  $p_{SRS-L}^{(k)}$ 에 따라 링크 서비스 벡터  $\pi^{(k)}$ 를 선택하고,  $\pi_l^{(k)} = 1$ 인 링크  $l$ 들을 활성화한다.

2) 활성화 세션 선택: 활성화된 링크  $l$ 은 FIFO (First-In-First-Out) 방식에 따라 큐의 맨 앞에 위치한 패킷을 전송한다.

SRS-L 스케줄러에서 링크  $l$ 의 활성화 확률이  $\lambda'_l + \epsilon_{SRS-L}$ 임을 쉽게 알 수 있으며 이 값은 명목상 패킷 도착률  $\lambda'_l$ 보다 크다. 서비스율에 대한 명목상 패킷 도착률의 비율을 명목상 부하라고 정의하자. SRS-S와는 달리 서로 다른 세션이 동일한 큐를 공유할 때는 명목상 부하가 1보다 작다고 해서 네트워크가 안정적인 것은 아니다. 모든 링크에서 명목상 부하가 1보다 작음에도 불구하고 큐가 무한히 증가하는 네트워크의 예들이 많이 알려져 있으며, 여기에서는 Rybko에 의해 제시된 간단한 유선 네트워크를 살펴보자<sup>[9]</sup>.

이 네트워크는 그림1과 같이 두 개의 서버 A, B로 구성되어 있고 두 개의 세션 1, 2가 서로 반대 방향으로 흐른다. 즉, 세션 1의 패킷은 서버 A에서 B

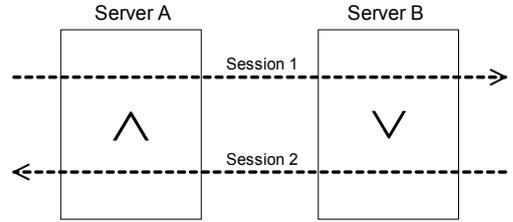


그림 1. 명목상 부하가 1보다 작지만 불안정한 네트워크의 예. 부등호는 서비스 우선순위를 가리킴.

로, 세션 2의 패킷은 서버 B에서 A로 향한다. 서버 A에서는 세션 2가 세션 1보다 높은 서비스 우선순위를 가지며 서버 B에서는 세션 1이 세션 2보다 높은 우선순위를 갖는다. 세션 1과 2의 패킷 도착률의 합은 1이라고 가정하자. 또한, 세션 1의 패킷이 경로상의  $i$ 번째 서버에서 겪는 평균 서비스 시간을  $m_i^1$  ( $i=1, 2$ ), 세션 2의 패킷이 겪는 서비스 시간을  $m_i^2$ 라고 표기하자. 각 서버에서의 평균 서비스 시간이  $m_1^1 + m_2^1 < 1$ ,  $m_2^2 + m_1^2 < 1$ 이면 두 서버에서 모두 명목상 부하가 1보다 작다. 그러나, 서비스 우선순위가 높은 세션들의 평균 서비스 시간 합이 1보다 큰 경우, 즉  $m_2^2 + m_2^1 > 1$ 인 경우 이 네트워크는 불안정하다고 알려져 있다.

직관적으로 그 이유를 살펴보면 다음과 같다. 일시적으로 서버 A에서 세션 2의 패킷이, 서버 B에서 세션 1의 패킷이 동시에 서비스되고 있다고 가정하자. 시간이 지나 서버 A에서 세션 2의 큐가 고갈되었다고 가정하자. 여전히 서버 B는 세션 1을 서비스하고 있으므로, 서버 A에는 더 이상 세션 2의 패킷이 도착하지 않는다. 그러다가, 서버 B에서 세션 1의 큐가 고갈되면 비로소 세션 2를 서비스하고, 이 때부터 서버 A에는 세션 2의 패킷이 다시 도착한다. 서버 A는 세션 2를 먼저 서비스하므로 세션 1의 패킷들은 더 이상 서버 B로 공급되지 않는다. 즉, 서버 A에서 세션 2가, 서버 B에서 세션 1이 동시에 서비스되는 경우는 발생하지 않으며, 따라서 서버 A에서의 세션 2와 서버 B에서의 세션 1은 마치 하나의 서버에 의해 서비스되는 것처럼 동작한다.  $m_2^2 + m_2^1 > 1$ 인 경우 그 가상 서버에서 큐가 무한히 증가하고 네트워크는 불안정하게 된다.

모든 링크에서 서비스율이 패킷 도착률보다 크다고 해서 네트워크가 안정적인지는 않으나, 적어도 유력한 조건이라는 것은 경험적으로 알려져 있다. 특히, 1) 각 링크에서의 평균 패킷 서비스 시간이 세션에 관계없

이 동일하고, 2) 모든 큐가 FIFO 방식으로 서비스된다는 조건이 추가되면, 모든 링크에서 명목상 부하가 1보다 작은 네트워크는 안정적이라고 증명되어 있다<sup>19)</sup>. 따라서, SRS-L에서 선택된 링크 큐를 FIFO 방식으로 서비스하는 것은 네트워크의 안정성을 위해 필수적이다.

이제 SRS-L 스케줄링에 대한 DTMC 모델을 수립하자. 슬롯  $n$ 에서 네트워크 내의 모든 큐 길이를 벡터  $\mathbf{Q}_{SRS-L}[n] := (Q_l[n])$ 으로 나타내자. SRS-L은 세션 큐를 유지하지 않지만 표기의 편의상, 슬롯  $n$ 에서 세션  $s$ 의 경로 상에 있는  $i$ 번째 링크가 활성화되고 이때 세션  $s$ 의 패킷이 전송되면  $\mu_i^s[n] = 1$ , 그렇지 않으면  $\mu_i^s[n] = 0$ 이라고 정의하자. 링크 큐  $Q_l$ 의 길이는 시간에 따라

$$Q_l[n+1] = Q_l[n] + \sum_{s|(s,i) \in \mathcal{S}} \mu_i^s[n] - \mu_l[n]$$

로 변화한다. 이 때, 정의에 따라

$\mu_l[n] = \sum_{(s,i) \in \mathcal{S}} \mu_i^s[n]$  임은 자명하다. 다음 슬롯에서의 큐 길이  $Q_{SRS-L}[n+1]$ 는 현재 슬롯에서의 정보  $Q_{SRS-L}[n]$ ,  $A^s[n]$ ,  $\mu_i^s[n]$ 에 의해서 결정되므로,  $Q_{SRS-L}[n]$ 은 DTMC이다.

다음의 명제는 Fluid Limit 이론<sup>19)</sup>에 의해 증명할 수 있으나 지면 관계상 생략한다. 본 논문의 분석 방법은 DTMC가 정상 상태를 가질 때에만 적용될 수 있으므로 명제 5는 SRS-L의 분석을 위해 반드시 필요하다.

**명제5:** SRS-L 스케줄링에 대한 DTMC  $Q_{SRS-L}[n]$ 은 양의 회귀성을 가지며, 따라서 정상 확률분포가 존재한다.

#### 4.2 패킷 지연

SRS-L의 패킷 지연을 구하는 과정은 SRS-S의 경우와 동일하다. 즉, DTMC  $Q_{SRS-L}[n]$ 의 Lyapunov 함수를 정의하고 정상 상태에서 그 변화율이 0임을 이용하여 평균 큐의 길이를 얻은 후 Little의 정리로부터 패킷 지연을 구한다. 이 과정에서 큐 길이와 큐 서비스 항을 분리할 필요가 있으며 이 때 다음의 명제 6이 적용된다. 이 명제는 SRS-S에서의 명제 3과 유사하며, 같은 방식으로 증명할 수 있다.

**명제6:**  $\vec{\lambda} \in A$ 일 때, SRS-L 스케줄링에 대해 다음의 두 식이 성립한다.

- 1)  $E\{Q_l[n] \mu_{i-1}^s[n]\} = E\{Q_l[n] \lambda^s\}$ ,
- 2)  $E\{Q_l[n] \mu_l[n]\} = E\{Q_l[n] (\lambda_l' + \epsilon_{SRS-L})\}$ .

명제 2와 6으로부터 이번 장의 주요 결과인 다음의 정리를 증명할 수 있으며, 패킷 지연을 간단히 나타내기 위해 각 노드에서 라디오 인터페이스는 1개라고 가정한다.

**정리2:**  $\vec{\lambda} \in A$ 일 때, SRS-L 스케줄러의 평균 패킷 지연은

$$\frac{\sum_s (\sigma^2(A^s) + (2H^s - 1)\lambda^s - (\lambda^s)^2)}{2\epsilon_{SRS-L}\lambda_{tot}} - \frac{\sum_l \sum_{(s,1),(t,1) \in \mathcal{S}, s \neq t} \lambda^s \lambda^t}{2\epsilon_{SRS-L}\lambda_{tot}}$$

이며,  $\widehat{\epsilon}_{SRS-L} := \sup \epsilon_{SRS-L}$ 에서 최소화된다.

**증명:** DTMC  $Q_{SRS-L}[n]$ 의 Lyapunov 함수를  $V(Q_{SRS-L}[n]) := 2^{-1} \sum_l (Q_l[n])^2$ 로 정의하면,  $V(\cdot)$ 의 변화율은 다음과 같다.

$$\begin{aligned} E\{V(Q_{SRS-L}[n+1]) - V(Q_{SRS-L}[n])\} &= \sum_l E\{Q_l[n] (\sum_{(s,i) \in \mathcal{S}} \mu_i^s[n] - \mu_l[n])\} \\ &\quad + c_{SRS-L} \\ &= -\epsilon_{SRS-L} \sum_l E\{Q_l[n]\} + c_{SRS-L} \end{aligned}$$

위 식에서 마지막 등식은 명제 6과 링크  $l$ 의 패킷 도착률  $\lambda_l = \sum_{(s,i) \in \mathcal{S}} E\{\mu_i^s[n]\}$ 로부터 따른다.

$c_{SRS-L} := 2^{-1} \sum_l E\{(\sum_{(s,i) \in \mathcal{S}} \mu_i^s[n] - \mu_l[n])^2\}$ 로 정의되었으며, 각 노드에서 라디오 인터페이스가 1개인 경우 아래와 같이 정리할 수 있다. 표기의 간결함을 위해 시간을 가리키는 인덱스  $n$ 은 생략한다.

먼저, 링크  $l$ 로의 외부 입력  $A_l := \sum_{(s,1) \in \mathcal{S}} \mu_0^s$ 와 네트워크 내부 입력  $I_l := \sum_{(s,i) \in \mathcal{S}} \mu_i^s - A_l$ 을 정의하자. 1) 노드 당 인터페이스가 1개인 경우  $\mu_l I_l = 0$ 이고, 2)  $(I_l)^2 = I_l$ ,  $(\mu_l)^2 = \mu_l$ 이며, 3)  $E A_l + E I_l = \lambda_l$ 이므로

$$\begin{aligned} c_{SRS-L} &= 2^{-1} \sum_l E\{(A_l + I_l)^2 - 2\mu_l(A_l + I_l) + (\mu_l)^2\} \\ &= 2^{-1} \sum_l E\{(A_l)^2 + 2A_l I_l + I_l - 2\mu_l A_l + \mu_l\} \\ &= 2^{-1} \sum_l (E(A_l)^2 - 2(E A_l)^2 + E I_l + \mu_l) \end{aligned}$$

로 쓸 수 있다. 두 번째 등식에서 1)과 2), 세 번째 등식에서 3)이 각각 사용되었다. 또한, 3)과  $\sum_l \mathbf{E}A_l = \sum_s \lambda^s$ 을 이용하면

$$\begin{aligned} \sum_l \mathbf{E}(I_l + \mu_l) &= \sum_l (2\lambda_l - \mathbf{E}A_l) \\ &= \sum_s (2H^s - 1)\lambda^s \end{aligned}$$

가 성립함을 알 수 있으며, 간단한 계산에 의해

$$\begin{aligned} \sum_l (\mathbf{E}(A_l)^2 - 2(\mathbf{E}A_l)^2) &= \sum_s (\sigma^2(A^s) - (\lambda^s)^2) \\ &\quad - \sum_l \sum_{(s,1),(t,1) \in \mathbf{g}, s \neq t} \lambda^s \lambda^t \end{aligned}$$

임을 확인할 수 있다. 따라서,

$$\begin{aligned} c_{SRS-L} &= 2^{-1} \sum_s (\sigma^2(A^s) + (2H^s - 1)\lambda^s - (\lambda^s)^2) \\ &\quad - 2^{-1} \sum_l \sum_{(s,1),(t,1) \in \mathbf{g}, s \neq t} \lambda^s \lambda^t \end{aligned}$$

로 정리된다.

명제 1에 의해  $V(\cdot)$ 의 변화율은 0이므로 네트워크 내의 모든 큐 길이의 평균은  $\mathbf{E}Q_{tot}[n] = \epsilon_{SRS-L}^{-1} c_{SRS-L}$ 이며, Little의 정리에 의해 평균 패킷 지연은  $(\lambda_{tot} \epsilon_{SRS-L})^{-1} c_{SRS-L}$ 로 주어진다. 패킷 지연은  $\epsilon_{SRS-L}$ 에 따라 단조 감소하므로 최소의 지연을 얻기 위해서는 가능한 큰  $\epsilon_{SRS-L}$ 을 선택해야 한다.

### 4.3 SRS-S와 SRS-L의 패킷 지연 비교

SRS-S와 SRS-L의 분석 결과로부터 큐의 결합이 평균 패킷 지연에 미치는 영향을 살펴보자. 정리 1로부터 SRS-S의 최소 평균 패킷 지연은

$$\frac{\sum_s (\sigma^2(A^s) + (2H^s - 1)\lambda^s - (\lambda^s)^2)}{2\epsilon_{SRS-S} \lambda_{tot}}$$

이며, 정리 2로부터 SRS-L의 최소 평균 패킷 지연은

$$\frac{\sum_s (\sigma^2(A^s) + (2H^s - 1)\lambda^s - (\lambda^s)^2)}{2\epsilon_{SRS-L} \lambda_{tot} - \frac{\sum_l \sum_{(s,1),(t,1) \in \mathbf{g}, s \neq t} \lambda^s \lambda^t}{2\epsilon_{SRS-L} \lambda_{tot}}}$$

이다. 위의 패킷 지연은 각 노드의 라디오 인터페이스 수가 1이라는 가정 하에서 성립하지만, 큐 결합에 의한 평균 지연의 영향을 살펴보는 데에는 충분하다.

### 1) SRS-L에서

$$(2\hat{\epsilon}_{SRS-L} \lambda_{tot})^{-1} \sum_l \sum_{(s,1),(t,1) \in \mathbf{g}, s \neq t} \lambda^s \lambda^t$$

는 서로 다른 세션의 첫 번째 링크가 동일할 때 그 세션들의 패킷 도착 프로세스 사이의 상관관계로부터 발생하며, 각 세션이 모두 다른 링크에서 시작하는 경우 0이다. 이 항은 항상 0보다 크거나 같으므로 SRS-L의 패킷 지연을 감소시킨다.

### 2) $\hat{\epsilon}_{SRS-S}$ 와 $\hat{\epsilon}_{SRS-L}$ 는 각각

$$\vec{\lambda}' + \epsilon_{SRS-S} (\mathbf{S}_1, \dots, \mathbf{S}_L) \in \Pi,$$

$\vec{\lambda}' + \hat{\epsilon}_{SRS-L} (1, \dots, 1) \in \Pi$ 를 만족하는 최대의 실수이다. 모든 링크  $l$ 에 대해  $|\mathbf{S}_l| \geq 1$ 이고  $\Pi$ 는 컨벡스 집합이므로

$$\hat{\epsilon}_{SRS-L} \geq \hat{\epsilon}_{SRS-S} \cdot \min |\mathbf{S}_l| \geq \hat{\epsilon}_{SRS-S}$$

임을 알 수 있다.

1)과 2)에 의해 SRS-L은 SRS-S보다 평균 패킷 지연이 작으며, 지연의 차이는 같은 링크에서 시작하는 세션의 수 및 패킷 도착률, 그리고 같은 링크를 경유하는 세션의 수에 의해 결정된다.

## V. 모의실험 및 수치분석

이번 장에서는 간단한 예를 통해 SRS-S와 SRS-L의 패킷 지연을 살펴보자. 고려하는 네트워크 토폴로지는 그림 2와 같은 덤벨(dumbbell)과 트리(tree)이며, 각각의 경우에 대해 점선으로 표시된 경로를 따라 이동하는 4개의 세션이 있다. 주어진 토폴로지에서 세션들의 패킷 도착률은 동일하며, 패킷의 입력은 Bernoulli 프로세스를 따른다.

그림 3은 패킷 도착률에 따른 평균 전송 지연을 로그 스케일로 나타내고 있다. 수치분석은 정리 1과 2에서 제시된 수식에 따라 계산되었으며, 분석의 정확성을 보이기 위해 모의실험이 병행되었다. 모의실험은 한 포인트 당  $10^8$  슬롯 동안 수행되었으며, 그림에서 “sim”으로 표기되었다. 또한, SRS-S의 경우<sup>[11]</sup>에서 제

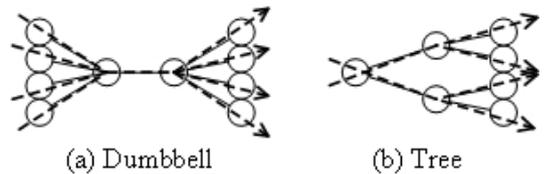


그림 2. 고려하는 네트워크 토폴로지.

시된 상계도 함께 나타내었다.

덤벨 토폴로지의 최대 네트워크 용량은 0.125이며 그림 3에서 점선으로 표시되어 있다. 덤벨의 경우 같은 링크에서 시작하는 세션들은 없지만 모든 세션이 가운데에 위치한 링크를 공유하므로 SRS-L이 SRS-S 보다 약 40% 작은 지연을 보인다. 트리 토폴로지의 최대 네트워크 용량은 0.25이며 역시 점선으로 표시되어 있다. 트리의 경우 두 쌍의 세션이 동일한 링크에서 시작하지만 라우팅 경로를 공유하는 세션이 2개 씩으로 덤벨의 경우보다 작으며, 이로 인해 SRS-L의 지연은 SRS-S에 비해 약 25%만 작다. 즉, 두 스케줄링의 지연 성능을 차별화하는 가장 중요한 요소는 동일한 링크를 공유하는 세션의 수임을 알 수 있다.

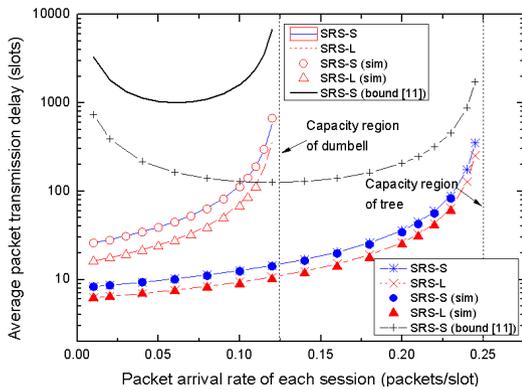


그림 3. SRS-S와 SRS-L의 평균 패킷 전송 지연.

## VI. 결 론

본 연구에서는 멀티홉 무선 네트워크에서 랜덤 스케줄링을 사용할 때 동일 링크 상의 세션 큐를 결합함으로써 얻을 수 있는 패킷 지연의 변화를 정량적으로 살펴보았다. 링크 기반의 랜덤 스케줄링은

네트워크 용량 내의 임의의 트래픽을 수용할 수 있으며 각 링크에서 큐 관리의 부담이 적을 뿐만 아니라, 평균 전송 지연의 측면에서도 주목할 만한 이득이 있음을 확인하였다.

## 참 고 문 헌

[1] P. Gupta and P. R. Kumar, "The capacity of wireless networks," *IEEE Trans. Information Theory*, Vol.46, No.2, pp.388-404, Mar. 2000.  
 [2] X. Lin, N. Shroff, and R. Srikant, "A tutorial on cross-layer optimization in wireless net-

works," *IEEE Journal on Selected Areas in Commun.*, Vol.24, No.8, pp.1452-1463, Aug. 2006.  
 [3] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," *IEEE Trans. Automatic Control*, Vol.37, No.12, pp.1936-1948, Dec. 1992.  
 [4] X. Lin and N. Shroff, "The impact of imperfect scheduling on cross-layer rate control in wireless networks," *IEEE INFOCOM*, 2005.  
 [5] A. Eryilmaz, A. Ozdaglar and E. Modiano, "Polynomial complexity algorithms for full utilization of multi-hop wireless networks," *IEEE INFOCOM*, 2007.  
 [6] L. Jiang and J. Walrand, "A distributed CSMA algorithm for throughput and utility maximization in wireless networks," *Annual Allerton Conference on Communication, Control, and Computing*, 2008.  
 [7] M. Neely, E. Modiano, and C. Rohrs, "Dynamic power allocation and routing for time varying wireless networks," *IEEE INFOCOM*, 2003.  
 [8] G. Gagan, and N. Shroff, "Delay analysis for multi-hop wireless networks," *IEEE INFOCOM*, 2009.  
 [9] M. Bramson, "Stability of queueing networks," *Probability Surveys*, Vol.5, pp.169-345, 2008.  
 [10] H. K. Khalil, *Nonlinear Systems*, Prentice Hall, 2002.  
 [11] L. Le, K. Jagannathan, and E. Modiano, "Delay analysis of maximum weight scheduling in wireless ad hoc networks," *Conference on Information Sciences and Systems (CISS)*, 2009.

최진구 (Jin-Ghoo Choi)

중신회원



1998년 2월 서울대학교 전기공학부  
 2000년 2월 서울대학교 전기공학부 석사  
 2005년 2월 서울대학교 전기컴퓨터공학부 박사  
 2010년 3월~현재 영남대학교 정보통신공학과 전임강사

<관심분야> 이동 통신망, 센서 네트워크