2020/07/22 @ 대양 AI 센터 736, 세종대학교

대기행렬의 기초

- 대기행렬의 개념-

권 순 홍 (soonhong@pel.sejong.ac.kr)

세종대학교 프로토콜공학연구실

목 차

대기행렬의 개념

- 대기행렬 (Queue)
  - 정의
    - 기회를 기다리는 행렬
    - 서비스를 제공받기 위해 기다리는 행렬
  - 형성 이유
    - 수요보다 공급이 적은 물리계의 가장 기본적인 형상에서 비롯됨
    - 서비스에 대한 수요와 서비스를 제공하는 공급 시스템의 능력 사이에 존재하는 일시적인 불균형

대기행렬의 개념

- 대기행렬
  - 대기행렬에 대한 대안
    - 수요보다 공급을 크게 할 경우
      - 이론적으로 대기행렬을 해결할 수 있는 방안
        - e.g., 사람이 필요로 하는 물건의 양보다 더 많은 양을 생성하여 공급
  - 대기행렬에 대한 한계
    - 수요에 대한 정확한 예측이 어려움
      - 사람이 필요로 하는 양이 시시각각 변화함
    - 수요를 정확히 예측하더라도 더 많은 공급에 있어 비용 증가
      - 일반적으로 수요에 대한 평균 값을 구하여 공급량을 결정
        - 예측했던 수요와 비교하여 더 큰 수요가 발생할 경우가 존재

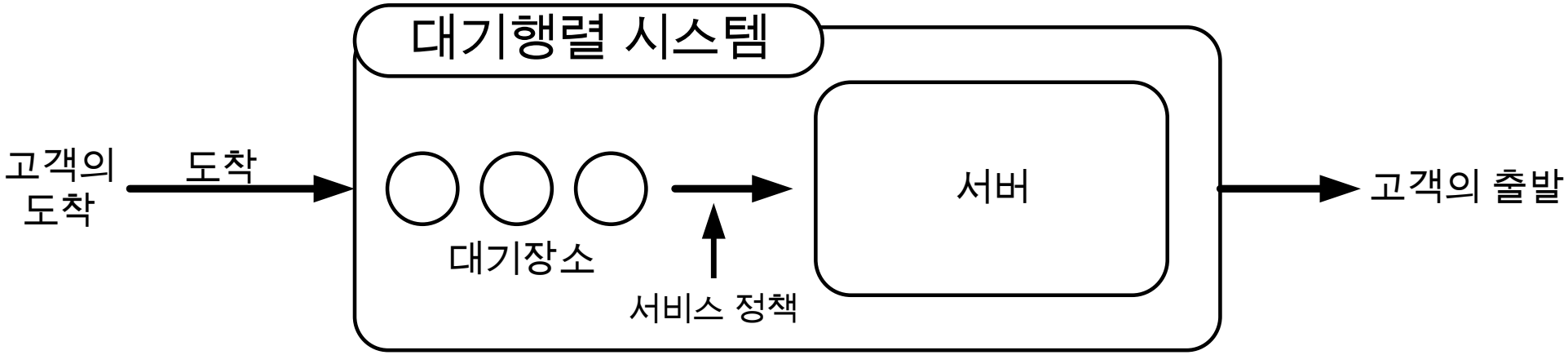
대기행렬의 개념

- 대기행렬 이론 (Queueing theory)
  - 덴마크 수학자 얼랑 (A. K. Erlang)이 1913년 최적의 전화교환기 수를 결정하기 위해 개발한 기법
  - 대기행렬의 특성과 현상을 수학적으로 분석하는 기법

  - 대기행렬이 이용되는 분야
    - 통신망 공학분야
    - 웹서버
    - 공장 자동화 시스템
    - 도로 교통 분야
    - 비즈니스 분야

대기행렬의 개념

- 대기행렬 구성요소
  - 대기행렬 시스템은 크게 고객의 도착, 대기장소, 서버, 고객의 출발의 네 부분으로 구성



대기행렬의 개념

- 대기행렬 구성요소
  - 고객의 도착
    - 장소나 시간 등 다양한 이유에 의해 다양한 형태를 지님
    - 고객의 규모
      - 유한고객 (Finite calling unit)
      - 무한고객 (Infinite calling unit)
    - 고객의 도착 유형
      - 고객의 도착간격 시간 분포
      - 한번에 도착할 수 있는 고객의 규모

대기행렬의 개념

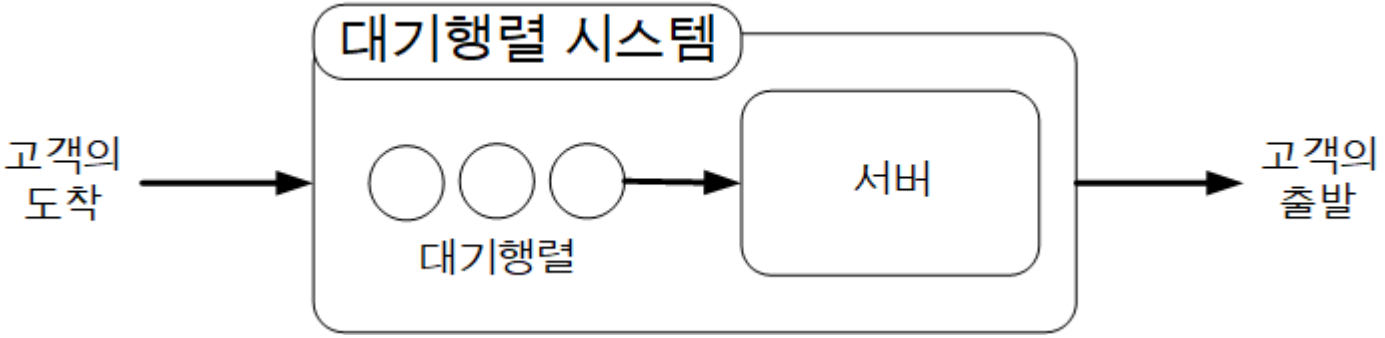
- 대기행렬 구성요소
  - 대기장소
    - 특정 이벤트에 대해 대기장소의 규모에 따라 대기행렬 형성
    - 주말의 기차역 등 특정한 이벤트가 있는 곳은 flash crowd들로 북적일 수 있음
      - 한정된 공간내에 들어갈 수 있는 인원 수가 정해져 있어 오버플로우 발생
    - 인터넷 라우터에 어느 순간 너무 많은 패킷이 들어올 경우, 버퍼 오버플로우 발생 가능
      - 다량의 패킷으로 인해 라우터 버퍼의 용량을 초과
      - 버퍼 오버플로우 발생시, 패킷이 소실될 가능성이 있음

대기행렬의 개념

- 대기행렬 구성요소
  - 서버
    - 서비스를 제공해주는 모든 장치
    - 버스의 경우 고객이 버스를 타는 것을 서비스로서 설명할 수 있음
      - 버스를 탈 때, 모든 사람이 자리에 앉은 후 버스가 출발한다면 이때 걸리는 시간을 서비스 시간이라고 함
    - 서버는 단일 서버 또는 다중 서버로 구성됨
      - 버스의 경우 한번에 여러 명의 사람을 태울 수 있으므로 다중 서버
      - 의사가 한 명인 병원의 경우, 병원의 서버는 단일 서버

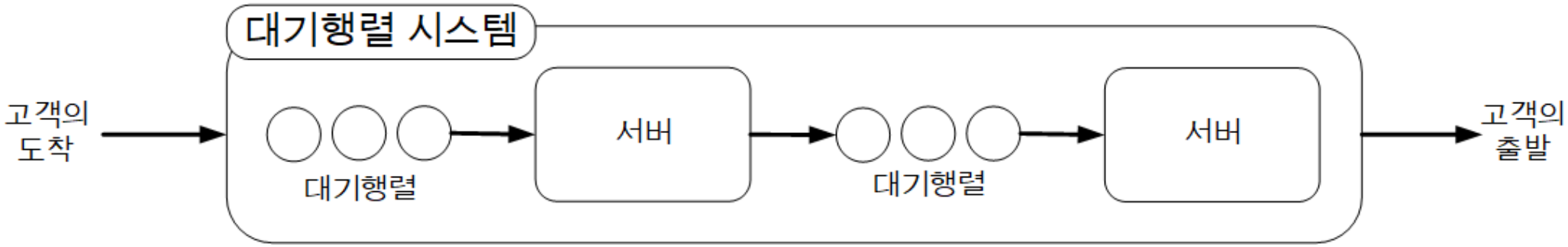
대기행렬의 개념

- 대기행렬 구성요소
  - 서버의 특성에 다른 대기행렬 구조
    - 단일경로-단일단계 (Single channel, single phase)



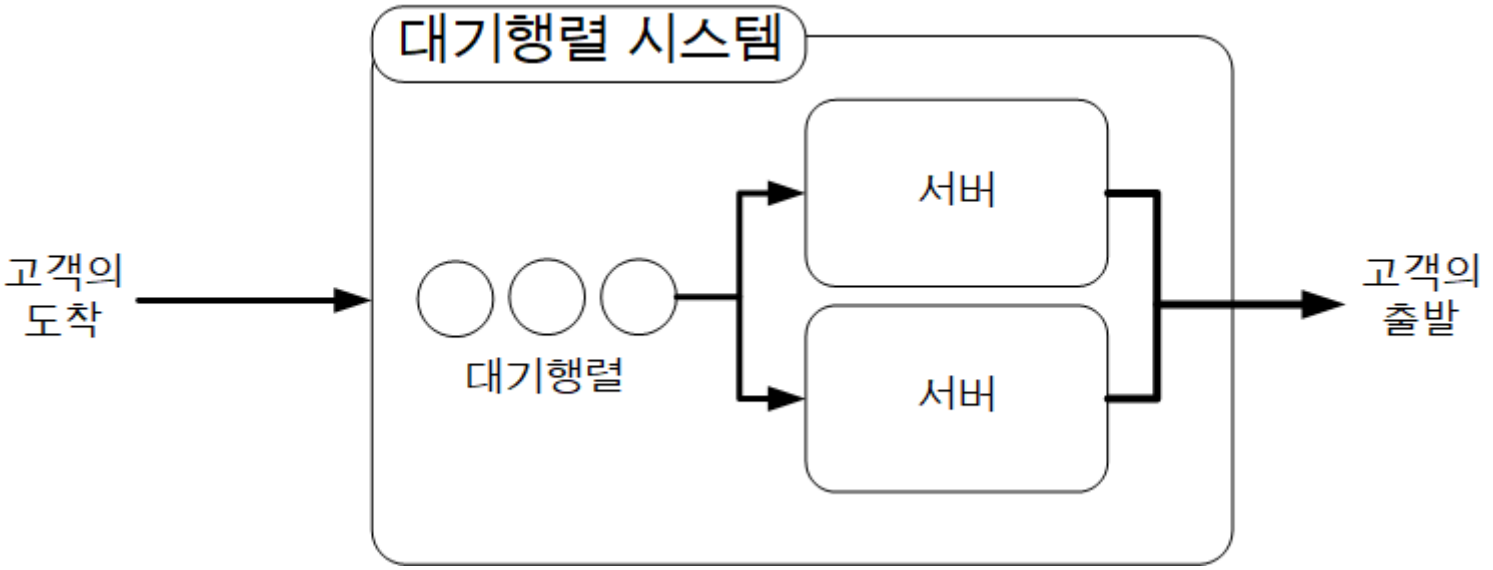
대기행렬의 개념

- 대기행렬 구성요소
  - 서버의 특성에 다른 대기행렬 구조
    - 단일경로-다수단계 (Single channel, multi phase)



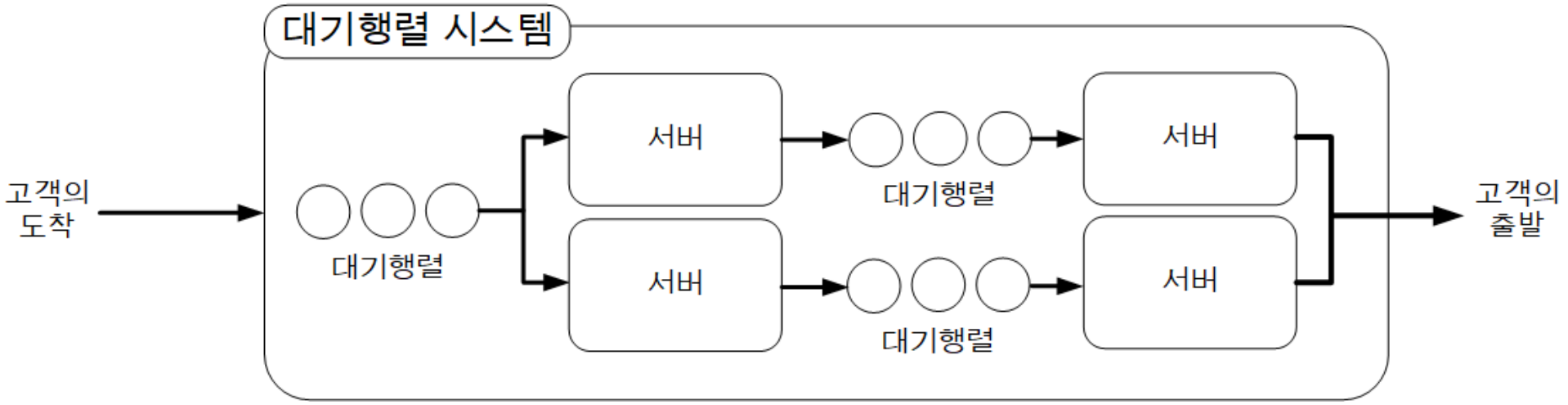
대기행렬의 개념

- 대기행렬 구성요소
  - 서버의 특성에 다른 대기행렬 구조
    - 다수경로-단일단계 (Multi channel, single phase)



대기행렬의 개념

- 대기행렬 구성요소
  - 서버의 특성에 다른 대기행렬 구조
    - 다수경로-다수단계 (Multi channel, multi phase)



대기행렬의 개념

- 대기행렬의 표기
  - 켄달의 대기행렬 모형 표시 방법
    - 1953년 켄달에 의해 5가지 항목을 기초로 하는 대기행렬 모형을 구분할 수 있는 새로운 표기법 개발
      - *A* / *B* / *c* / *K* / *P*로 표현
        - *A* : 고객의 도착에 관한 정보
        - *B* : 고객의 서비스 시간 분포
        - *c* : 시스템이 가지고 있는 서버의 수
        - *K* : 시스템이 수용할 수 있는 고객의 수
        - *P* : 고객의 규모

대기행렬의 개념

- 대기행렬의 표기
  - 켄달의 대기행렬 모형 표시 방법
    - *A* / *B* / *c* / *K* / *P*
      - 고객의 도착에 관한 정보를 나타냄
      - 인접해서 도착하는 고객 간의 시간간격이 어떠한 분포를 따르는지 나타냄
        - e.g., 고객이 고정된 시간간격을 가지고 한명씩 규칙적으로 들어오는 경우, 아무런 연관성 없이 임의의 간격으로 들어오는 경우, 등

대기행렬의 개념

- 대기행렬의 표기
  - 켄달의 대기행렬 모형 표시 방법
    - *A* / *B* / *c* / *K* / *P*
      - 고객의 서비스 시간 분포를 나타냄

        - 서비스 시간이 균일한 경우
          - 동전을 넣고 공을 치는 야구장의 경우, 야구공이 나오는 간격은 거의 동일

        - 서비스 시간이 균일하지 않은 경우
          - 병원의 경우, 환자의 상태에 따라 의사가 진찰하는 시간이 균일하지 않음

대기행렬의 개념

- 대기행렬의 표기
  - 켄달의 대기행렬 모형 표시 방법
    - *A* / *B* / *c* / *K* / *P*
      - *A* 와 *B* 항목에 사용 될 수 있는 분포

        - *M* (Memoryless or Markov)
          - 고객의 도착간격이 지수분포를 나타냄

        - E_k (Erlang)
          - 이 분포함수는 고객의 도착간격을 *k* 차 감마분포로 나타냄

        - *D* (Deterministic)
          - 이 분포함수는 고객의 도착간격이 고정간격을 가짐

        - *G* (or *GI*) (General 또는 General Independent)
          - 일반 혹은 임의분포를 나타냄
          - 어떠한 분포도 포함할 수 있는 일반적 분포

대기행렬의 개념

- 대기행렬의 표기
  - 켄달의 대기행렬 모형 표시 방법
    - *A / B / c / K / P*
      - 시스템이 가지고 있는 서버의 수

        - 시스템이 단일 서버로 이루어져 있는 경우
          - 병원에 의사가 한 명일 경우, 한번에 진료를 받을 수 있는 환자의 수는 한 명이며, 서버의 수는 1이됨

        - 시스템이 여러개의 서버로 이루어져 있는 경우
          - 병원에 의사가 세 명일 경우, 동시에 진료를 받을 수 있는 환자의 수는 세 명이며, 서버의 수는 3이 됨

대기행렬의 개념

- 대기행렬의 표기
  - 켄달의 대기행렬 모형 표시 방법
    - *A* / *B* / *c* / *K* / *P*
      - 시스템이 수용할 수 있는 고객의 수를 나타냄

        - 버퍼의 크기가 *K* - 1이고 고객이 하나의 서버로부터 서비스를 제공 받고 있을 경우 (단일 서버가 하나의 고객을 처리하는 경우), 시스템이 수용할 수 있는 고객의 수 *K*
          - 예
            - 버퍼의 크기 (*K* - *1*) : 10
            - 서버의 크기 (*c*) : 1
            - 시스템이 수용 가능한 고객의 수 (*K*) : 11
        - 해당 항목이 생략되어 있는 경우에는 *K* 값이 무한대
        - 서버의 크기가 *c* 이면 *K* - *c* 명의 고객은 버퍼에서 기다려야함

대기행렬의 개념

- 대기행렬 모형의 대표적인 예
  - *M*/ *M*/ *1* 큐
    - 도착 과정 (*M*)
      - 고객의 도착간격이 지수함수적으로 분포하고 있음
      - 임의의 관측시간 동안 도착한 고객의 수가 푸아송 분포 함수로 표시됨
    - 서비스 과정 (*M*)
      - 고객의 서비스 시간이 지수함수 분포를 따름
    - 서버의 수 (*1*)
      - 단일 서버로 구성되어 있음
    - 해당 대기행렬시스템의 해석에서 가장 관심을 가지는 것은 임의 시간 t에서 시스템 내에 있는 고객의 수 $X(t)$
      - BDP (Birth and Death Process)를 통해 구할 수 있음

대기행렬의 개념

- 대기행렬 모형의 대표적 예
  - $M/M/\infty$ 큐
    - 도착 과정 (M)
      - 고객의 도착간격이 지수함수적으로 분포되고 있음
      - 임의의 관측시간 동안 도착한 고객의 수가 푸아송 분포 함수로 표시됨
    - 서비스 과정 (M)
      - 고객의 서비스시간이 지수함수 분포를 따름
    - 서버의 수 (∞)
      - 서버의 수가 무한히 많은 서버

대기행렬의 개념

- 대기행렬 모형의 대표적 예
  - *M*/ *G*/ *1* 큐
    - 도착 과정 (*M*)
      - 고객의 도착간격이 지수함수적으로 분포하고 있음
      - 임의의 관측시간 동안 도착한 고객의 수가 푸아송 분포 함수로 표시됨
    - 서비스 과정 (*G*)
      - 고객의 서비스시간이 일반적인 함수분포를 따름
    - 서버의 수 (*1*)
      - 단일서버로 구성되어 있음
    - 해당 대기행렬시스템의 해석에서 가장 관심을 가지는 것은 정상 상태에서 시스템 내에 있는 고객의 수, 고객이 기다려야 하는 평균 지연시간 등이 있음
      - IMP (Imbedded Markov Process)의 해석 방법을 통해 구할 수 있음

대기행렬의 개념

- 대기행렬 모형의 대표적 예
  - 고객의 도착과정, 서비스과정, 서버의 수, 버퍼의 크기 등 크게 네 가지 요소에 의해 다양한 종류를 가짐
  - 서비스 과정에 있어 서비스 정책에 따라 여러 종류로 나뉨
    - FIFO (First In First Out)
      - 도착 순서에 따른 서비스 제공
    - Round Robin
      - 공평하게 배분되는 차례에 의한 서비스 제공
    - Strict Priority
      - 우선순위에 따른 차별화된 서비스 제공

리틀의 법칙

- 개요
  - MIT의 교수인 존 리틀이 1961년 Operation Research라는 저널에 발표한 이론
  - 대기행렬 이론의 해석에 있어 유용하게 사용되는 이론
  - 정상상태에서 대기행렬 시스템 내에 존재하는 고객의 평균 수
    - 고객의 평균 도착률 * 시스템 내 평균 체제시간
    - $L = \lambda E[W]$

리틀의 법칙

- 리틀의 법칙 표기법

표기법	설명
$Q(t)$	시간 t에서 시스템 내에 존재하는 고객의 수 (큐에서 기다리는 고객의 수와 서버에 의하여 서비스를 받고 있는 고객의 수의 합)
T	시스템 상태를 관찰자가 관측한 시간
L	시스템 내에 있는 평균 고객의 수
λ	고객의 시스템으로의 평균 도착률
W	임의의 한 고객의 시스템 내 체제시간 (큐에서 기다린 시간과 서버에 의해서 서비스를 받는 데 걸린 시간의 합)
$E[W]$	임의의 한 고객에 대한 시스템 내에서의 평균 체제시간
t_j	j번째 고객의 시스템 도착 시간
$C_1, C_2, \ldots$	시각 t_1, t_2, …에 도착하는 고객을 나타냄
$N[T]$	관측기간 T 동안 도착하는 고객의 수 $N(T) = \text{Max}\{ n : t_n \leq T \}$
d_j	C_j 의 출발시점
s_j	고객 j의 시스템 내 체제시간 $s_j = d_j - t_j$

리틀의 법칙

- 리틀의 법칙 단계별 도출

1. 처음 시스템 내에 고객이 없다고 가정
   - $Q(0) = 0$ 로 정의

2. 인식함수 정의 (I_j)
   - 임의의 고객이 시간 t 에서 시스템에 있는지 없는지를 나타내기 위함
     - C_j 가 시각 t 에 시스템 내에 존재할 경우
       - $I_j = 1$ 로 정의

리틀의 법칙

- 리틀의 법칙 단계별 도출

3. 시간 t에 시스템 내에 존재하는 고객의 수
   - 1번과 2번에서 정의한 수식을 기반으로 수식 도출
     - $Q(t) = \sum_{j=1}^{N(T)} I_j(t)$

4. 고객 j의 시스템 내 체제시간
   - $(t_j, t_j + s_j]$ 구간의 밖에 대해서는 $I_j(t) = 0$
     - $s_j = \int_0^T I_j(t)dt$

리틀의 법칙

- 리틀의 법칙 단계별 도출

5. 시스템 내에 있는 평균 고객의 수

- $L = \lim_{T\to\infty} \frac{1}{T}\int_0^T Q(t)dt$

$= \int_0^T Q(t)dt = \int_0^T \sum_{j=1}^{N(T)} I_j(t)dt$

$= \int_0^T \sum_{j=1}^{N(T)} I_j(t)dt = \sum_{j=1}^{N(T)} \int_0^T I_j(t)dt = s_1 + s_2 + \ldots + s_{N(T)}$

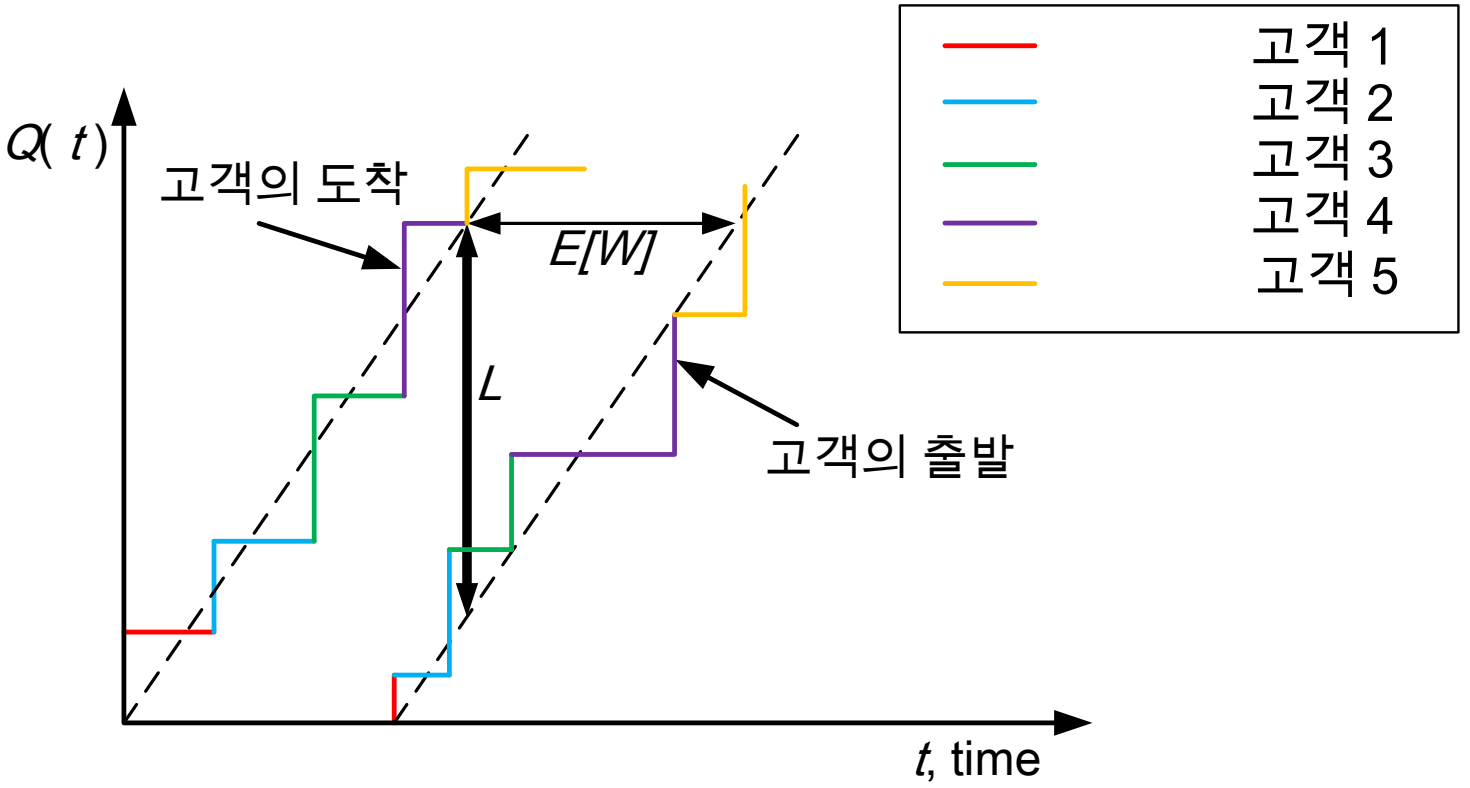
$= \lim_{T\to\infty} \frac{1}{T}\int_0^T Q(t)dt = \frac{s_1+s_2+\ldots+s_{N(T)}}{T}$

$= \frac{N(T)}{T}\frac{s_1+s_2+\ldots+s_{N(T)}}{N(T)}$

$= \lambda E[W]$

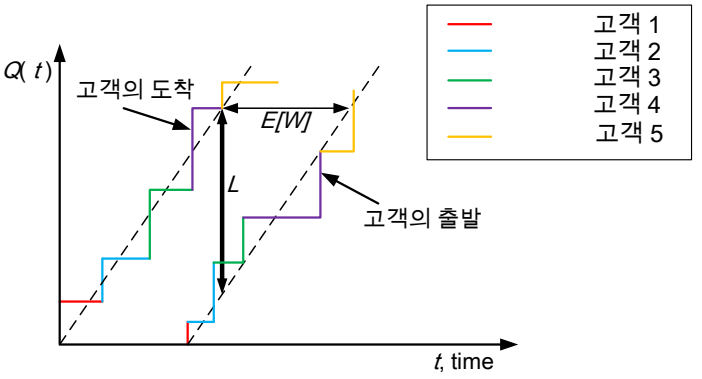
리틀의 법칙

- 시간의 흐름에 대한 시스템 내 고객의 수 변화
  - 시스템으로 들어온 고객에 대한 사건을 기록한 그래프
    - 안정적으로 동작하도록 설계된 시스템은 고객의 도착과 출발을 나타내는 기울기가 거의 같은 패턴을 이룸



리틀의 공식

- 시간의 흐름에 대한 시스템 내 고객의 수 변화
  - 시스템으로 들어온 고객에 대한 사건을 기록한 그래프
    - 충분히 긴 시간 T동안 그래프를 관측하면 다음의 결과를 확인 할 수 있음
      - T 동안 시스템으로 들어온 평균 고객의 수
        - λT
        - 단위시간 당 시스템으로 들어오는 고객의 수에 경과한 시간을 곱하여 얻음
      - T 동안 시스템을 나간 평균 고객의 수
        - $\lambda(T - E[W])$
        - T 동안 시스템으로 들어온 평균 고객의 수에서 그 동안 시스템에서 대기하고 있는 고객의 수를 제외한 고객의 수
      - T 동안 시스템 내에 남아 있는 고객의 수
        - $L = \lambda T - \lambda(T - E[W]) = \lambda E[W]$



리틀의 법칙

- Example 1

Q 1. 의사가 한명인 병원에 고객이 시간당 평균적으로 5명이 방문한다. 이 병원에는 환자가 병원에 도착하여 진료를 받고 병원을 나서기까지 평균적으로 30분은 소요된다고 한다. **그렇다면 진료를 받기 위하여 병원에 온 고객이 기다리는 동안 모두 앉아 있을 수 있게 하려면 최소한 몇 개의 의자가 필요한가 ?**

Answer:

고객의 평균도착률 λ가 5이고, 고객이 평균 병원에서 머무르는 평균 체재시간 $E[W]$ 이 0.5 (단위: 시간)이므로, 이 병원에 있는 평균 고객 수는 $L = \lambda E[W]$ = 5 × 0.5 = 2.5 이다.
고객의 수는 자연수로 나타내야 하므로 $\lfloor 2.5 \rfloor = 3$ 명이다. **따라서, 병원에는 최소 2개 이상의 의자가 준비 되어야 한다 (정상상태에서 병원 안에 평균 3명이 있으므로 한 명은 의사 앞에 앉아 진찰을 받고 있고 나머지 두 명은 대기실에 있어야 하기 때문이다.) .**

리틀의 법칙

- Example 2

Q 2. A 식당은 1시간에 평균적으로 20명의 손님이 방문한다. A 식당 안에서 식사하고 있는 손님은 평균 30명이다. **그렇다면 손님 한 명이 A 식당 안에서 머무는 시간을 얼마나 될까?**

Answer:

이는 리틀의 법칙을 이용하여 간단하게 구할 수 있다. $L = \lambda E[W]$ 해당 식에 위의 문제에서 나와있는 조건을 대입해 보면 다음과 같이 나타낼 수 있다. 식당 내 머무는 손님 (30명) = 방문하는 손님 (20명/시간) × 식당내 손님이 머무르는 시간 ($E[W]$) 를 통해 손님 한 명이 A 식당 안에서 머무르는 시간 $E[W]$ 를 구할 수 있다. **즉, 손님 한 명이 A 식당 안에서 머무르는 시간 $E[W]$ 는 1.5 시간인 것을 확인할 수 있다.**

리틀의 법칙

- Example 3

Q 3. 시간당 1,000,000건의 카드 승인을 처리하는 시스템이 있을 때, **초당 처리량에 대해 구하시오.**

Answer:

해당 문제의 경우, 시간당 1,000,000건의 카드 승인을 처리하는 시스템에 대해 초당 처리량에 대해 구하는 문제이다. 이는 리틀의 법칙을 이용하여 풀이가 가능하며, 처리량에 대한 수식은 다음과 같이 나타낼 수 있다. 처리량(TPS) = 서비스 처리 건수 / 측정 시간(초) . **즉, 해당 수식에 위의 문제에 나와있는 조건을 대입하면 처리량(TPS) = 1,000,000 / 3,600 으로 처리량은 278 TPS 이다.**

Thanks!

권 순 홍 (soonhong@pel.sejong.ac.kr)