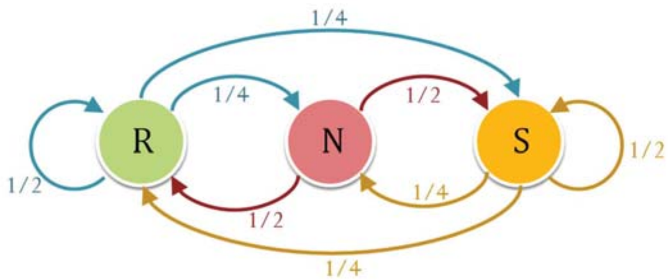
ECS455 Chapter 3

Call Blocking Probability



Dr.Prapun Suksompong
prapun.com/ecs455

Office Hours:
BKD, 6th floor of Sirindhralai building

Tuesday	14:20-15:20
Wednesday	14:20-15:20
Friday	9:15-10:15

Introduction

- The English dictionary word with the most consecutive vowels (six) is **EUOUAE**.
  - It is also the longest English word consisting only of vowels

- Imagine a word with **five** consecutive vowels.

Introduction

- Words with five consecutive vowels include **AIEEE**, C**OOEEI**NG, M**IAOUE**D, Z**AOUIA**, JUSS**IEUEA**N, Z**OOEAE**, Z**OAEAE**.

Can't find any of these words in Cambridge Dictionary

Introduction

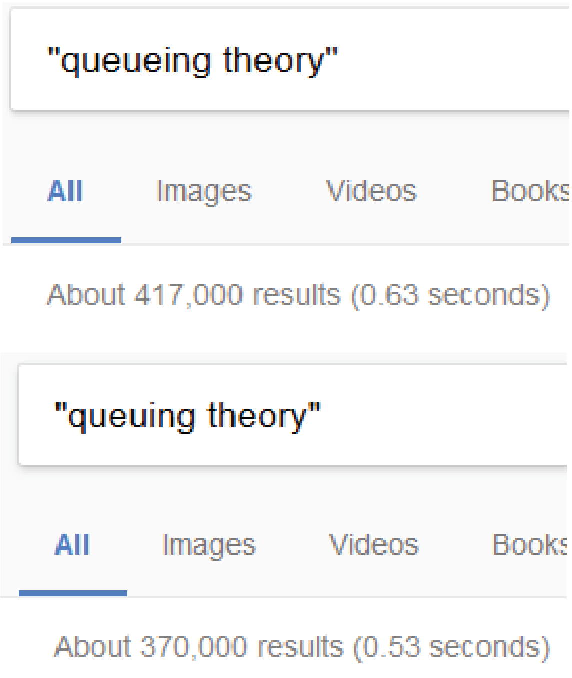
- Words with five consecutive vowels include **AIEEE**, C**OOEEI**NG, M**IAOUE**D, Z**AOUIA**, JUSS**IEUEA**N, Z**OOEAE**, Z**OAEAE**.

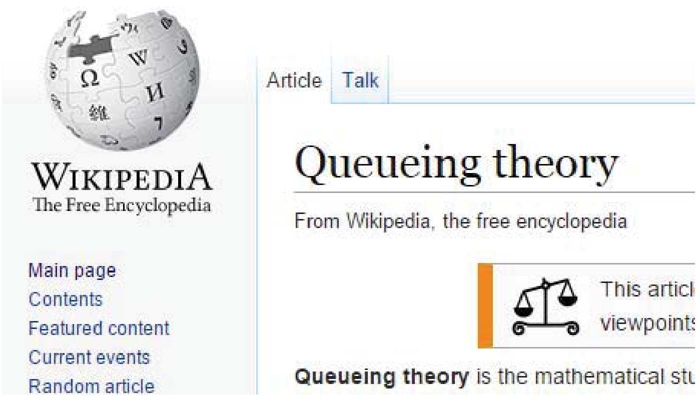
- Our new topic: Q**UEUEI**NG THEORY.
  - This is the only common word in the English language with five consecutive vowels.
- Note: The longest common word without any of the five vowels is RHYTHMS.
  - There are longer rare words: SYMPHYSY, NYMPHLY, GYPSYRY, GYPSYFY, and TWYNDYLLYNGS. WPPWRMWSTE and GLYCYRRHIZIN are long words with very few vowels.

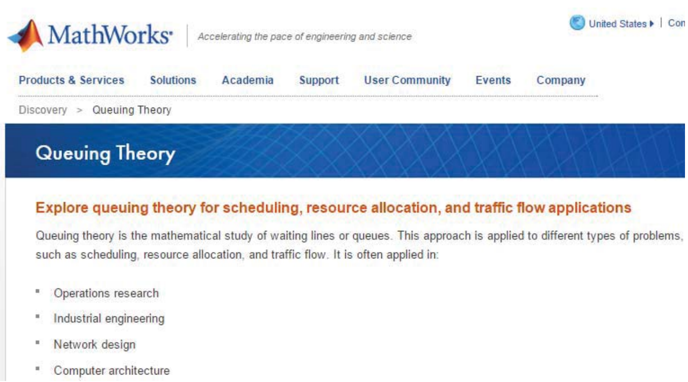
That Second "e"…

- You may recall the rule for changing a verb into its "–ing" form from your English class…
- If the verb ends in an "e" we remove the "e" and add "-ing":
  - browsing, causing, changing, charging, choosing, giving, having, hiring

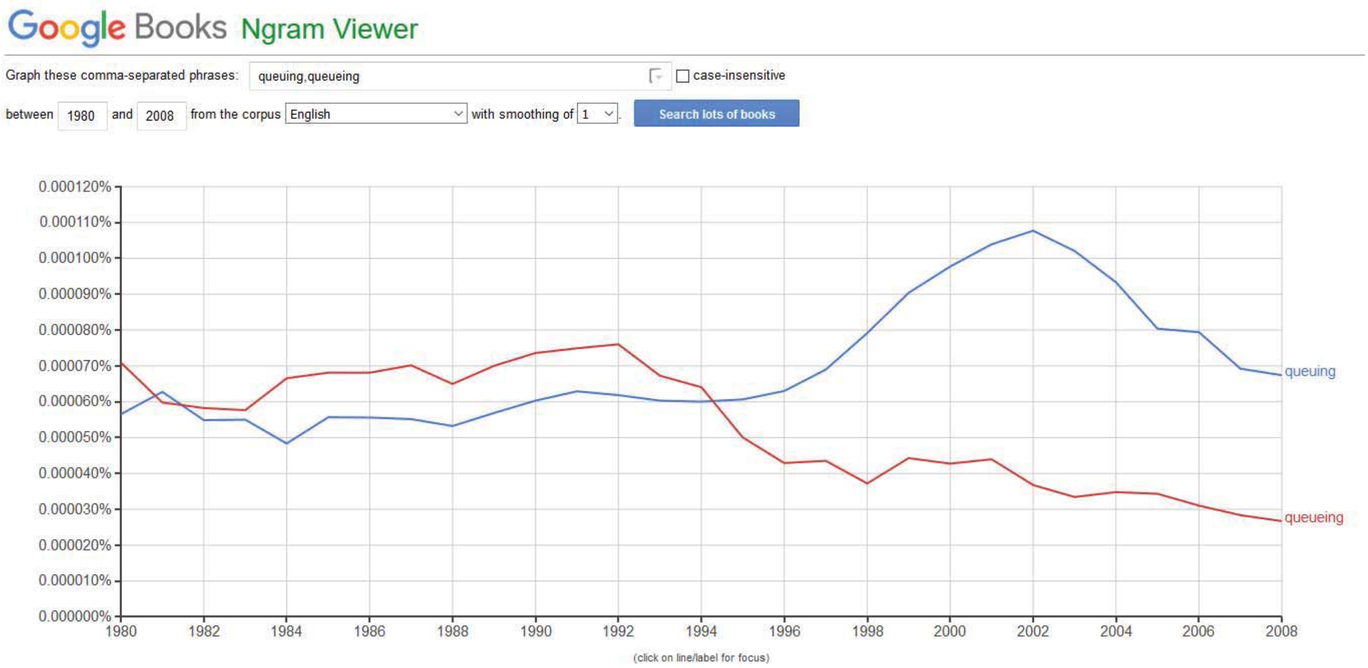
Queueing vs. Queuing







Queueing vs. Queuing



ECS455 Chapter 3

Call Blocking Probability

3.2 Markov Chain

Dr.Prapun Suksompong
prapun.com/ecs455

Review: Discrete-Time Markov Chain

- We model the evolution in time of K by Markov chain.
  - $K(t)$ = the number of channels being occupied at time t
- Time is divided into small slots so that our analysis can be done in discrete time.
  - This only approximate the solution. However, the answers will be accurate in the limit that the slot size δ approaches 0.
- Discrete-time Markov chain can be specified via its **state transition diagram** or its **probability transition matrix P**.

Simulating a Markov Chain in MATLAB

```
function X = MarkovChainGS(n,S,P,X1)
% n = the number of slots to be considered
% S = a row vector containing possible states (usually 1:N)
% P = transition probability matrix
% X1 = initial state for slot 1

N = length(S);      % Number of possible states
T = zeros(1,n);     % Preallocation
T(1) = find(S==X1); % Express the states using indices from 1 to N
                    % instead of the provided support S
for k = 2:n
    T(k) =  randsrc(1,1,[S;P(T(k-1),:)]);
end
X = S(T);           % Express the states using the provided support
end
```

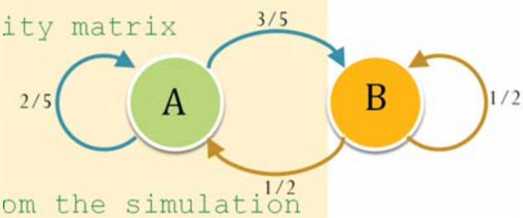
Simulating a Markov Chain in MATLAB

```
n = 1e1;                  % The number of slots to be considered
S = [1,2];                % Two possible states
P = [2/5 3/5; 1/2 1/2];  % Transition probability matrix
X1 = 2;                   % Initial state

X = MarkovChainGS(n,S,P,X1)

% Approximate the transition probabilities from the simulation
P_sim = []; x = X(1:(n-1)); y = X(2:n);
for k = 1:length(S)
    I = find(x==S(k)); LI = length(I);
    yc = y(I); cond_rel_freq = hist(yc,S)/LI;
    P_sim = [P_sim; cond_rel_freq];
end
P_sim

% Approximate the proportions of time that the states occur
p_sim = hist(X,S)./n
```



[MarkovChain_Demo1.m]

Example 3.10

```
>> MarkovChain_Demo1
X =
     2     2     1     1     1     2     2     1     2     1
P_sim =
    0.5000    0.5000
    0.6000    0.4000
p_sim =
    0.5000    0.5000
```

Exercises

```
>> MarkovChain_Demo1
X =
     2     1     2     1     2     1     2     1     1     1
P_sim =
    0.4000    0.6000
    1.0000         0
p_sim =
    0.6000    0.4000

>> MarkovChain_Demo1
X =
     2     2     2     2     1     2     2     2     1     1
P_sim =
    0.5000    0.5000
    0.2857    0.7143
p_sim =
    0.3000    0.7000
```

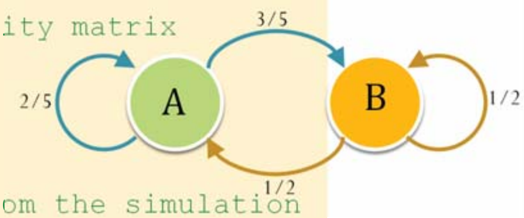

Example 3.7

```
n = 1e4;                    % The number of slots to be considered
S = [1,2];                  % Two possible states
P = [2/5 3/5; 1/2 1/2];     % Transition probability matrix
X1 = 2;                     % Initial state

X = MarkovChainGS(n,S,P,X1);

% Approximate the transition probabilities from the simulation
P_sim = []; x = X(1:(n-1)); y = X(2:n);
for k = 1:length(S)
    I = find(x==S(k)); LI = length(I);
    yc = y(I); cond_rel_freq = hist(yc,S)/LI;
    P_sim = [P_sim; cond_rel_freq];
end
P_sim

% Approximate the proportions of time that the states occur
p_sim = hist(X,S)./n
```




```
>> MarkovChain_Demo2
P_sim =
    0.4007    0.5993
    0.5055    0.4945
p_sim =
    0.4575    0.5425
```

[MarkovChain_Demo2.m]

Example 3.4

```
n = 1e4;                  % The number of slots to be considered
S = [1,2,3];              % Three possible states
P = [0 1 0; 0 2/3 1/3; 1/2 1/2 0]; % Transition probability matrix
X1 = 2;                   % Initial state

X = MarkovChainGS(n,S,P,X1);

% Approximate the transition probabilities from the simulation
P_sim = []; x = X(1:(n-1)); y = X(2:n);
for k = 1:length(S)
    I = find(x==S(k)); LI = length(I);
    yc = y(I); cond_rel_freq = hist(yc,S)/LI;
    P_sim = [P_sim; cond_rel_freq];
end
P_sim

% Approximate the proportions of time that the states occur
p_sim = hist(X,S)./n
```

```
>> MarkovChain_Demo3
P_sim =
         0    1.0000         0
         0    0.6620    0.3380
    0.4858    0.5142         0
p_sim =
    0.1093    0.6657    0.2250
```


Review: Call–Blocking Probability

- **Call blocking probability P_b** is the (long-term) proportion of calls that get blocked by the system because all channels are occupied.
- For M/M/m/m system, the (long-term) call blocking probability P_b is given by p_m

= the steady-state probability for state m

= the (long-term) proportion of time that the system will be in state m