Overview
- simple formulas, open or closed systems
- formalize intuition
- useful for:
  - system capacity planning, computing measures
e.g., given $X$ and $R$, compute $N$
  - verifying measured values
  - deriving new results & insights
- for open systems, assume: $X = \lambda$
  (i.e., $\lambda$ is not too high; queues large enough)

Outline
- Forced Flow Law
- Little's Result
  - Basic result
  - Example applications
- Verifying System Measures

Forced Flow Law: $X_k = V_k X$

- $X_k$ number of completions at node $k$
- $V_k$ length of measurement interval
- $X = \lambda$

Forced Flow Law – Numerical Example

- $S_1 = 0.6$
- $S_2 = 0.5$
- $S_3 = 1 \text{ sec}$
- $\lambda = 1/\text{sec}$
- $X_1 = X_2 = X$
- $\lambda = 1.5/\text{sec}$
- $X_1 = X_2 = X$

Lecture 5, ABA: simple formula to detect infeasible $\lambda$
Forced Flow Law – Numerical Example

\[ \lambda \]
\[ X_1 \]
\[ p=0.5 \]
\[ S_2=1.5 \text{ sec} \]
\[ 1-p \]
\[ S_3=1 \text{ sec} \]

\[ S_1 = 0.6 \]
\[ S_2 = 1.5 \text{ sec} \]
\[ S_3 = 1 \text{ sec} \]

\[ X \]

\[ \lambda = \text{1/sec} \]
\[ X_1 = \text{1/sec} \]
\[ X_2 = \text{0.5/sec} \]
\[ X = \text{1/sec} \]

\[ (a) \lambda = \text{1/sec} \]
\[ X_1 = \text{1/sec} \]
\[ X_2 = \text{0.5/sec} \]
\[ X = \text{1/sec} \]

\[ (b) \lambda = \text{1.5/sec} \]
\[ X_1 = \text{1.5/sec} \]
\[ X_2 = \text{0.67/sec} \]
\[ X = \text{1.42/sec} \]

\[ X \]

\[ \lambda \]
\[ V_1 \]
\[ P_{1,0} \]
\[ X_1 \]
\[ X_2 \]
\[ X_3 \]

\[ V_i = 5 \Rightarrow X_i = 5X \Rightarrow X = 0.2X_i \Rightarrow p_{1,0} = 0.2 \]

\[ \text{Conversely,} \]
\[ P_{1,0} = 0.2 \Rightarrow X = 0.2X_i \Rightarrow X_i = 5X \Rightarrow V_i = 5 \]

\[ \text{i.e.,} \]
\[ X = p_{1,0}X_1 \Rightarrow X_2 = (1/p_{1,0})X \text{ or } V_1 = 1/p_{1,0} \]

\[ X \]

\[ \lambda \]
\[ X \]
\[ V \]
\[ p_{1,0} \]

\[ X_2 = p_{1,2}X_1 \]
\[ X_1 = p_{1,2}V_1 \]
\[ X_2 = p_{1,2}V_1 \]

\[ \text{More generally,} \]
\[ X_i = \sum p_{i,j}X_j \text{ and } V_i = \sum p_{i,j}V_j \]

\[ \text{e.g.,} \]
\[ X_1 = X_0 + X_2 + X_3 \]
\[ V_1 = 1 + V_2 + V_3 \]

\[ X \]

\[ \lambda \]
\[ X_1 \]
\[ X_2 \]
\[ X_3 \]

\[ \gamma = \text{accumulated customer-seconds in } [0,T] \]
\[ = \text{area between the two curves} \]
\[ R = \text{average residence time in the "system"} \]
\[ N = \text{average number of customers in the "system"} \]

\[ \gamma = \text{XR} \]

\[ N = \text{XR} \]

\[ C = \text{average customer arrival rate} \]

\[ \gamma/C = \text{average throughput} \]

\[ \text{holds for open or closed system} \]

\[ \text{holds for open or closed system} \]
N = XR: Computing Measures

Application:
compute from NCSA O2K job trace:
X, average wait to run (W), average runtime (T)
use Little's result to estimate:
average number waiting to run = XW
average number waiting & running = X(W+T)
question: when are these estimates exact?

N = XR: Checking Measured Values

Application:
proxy server media caching policy simulation
average size of small files: 50 MB
large files: 500 MB
arrival rate of client requests: 15/min.
ratio of small to large requests: 1:4
file streaming rate = 4 Mbits/sec
total proxy disk bandwidth: 8 MBytes/sec
simulation result: hit rate = 0.33
average number of concurrent streams: 15/min×0.33×13.7 min

N = XR: types of applications

key types of applications:
compute measures, i.e.,
given two of the measures, compute the third
check the correctness of measured data
insight - e.g., SRPT scheduling minimizes N ⇒ min. R

Example: TCP Vegas

Algorithm for setting window (W)
- backlog = \( \frac{W}{RTT} \) (RTT - baseRTT)
- Slow Start: exit if backlog > \( \gamma \)
- every RTT in congestion avoidance:
  \[
  \begin{align*}
  W' &= W, \text{ if } \alpha < \text{backlog} < \beta \\
  W' &= W - W, \text{ if backlog} > \beta
  \end{align*}
  \]
(i.e., reduce W before loss occurs)

Summary

- Forced Flow Law, \( X_k = V_k X \)
  - \( x_k = \sum_{j=1}^{p} x_j \) and \( v_k = \sum_{j=1}^{p} v_j \)
- Little's Result, \( N = XR \)
- Key applications
  - System design & capacity planning
  - computing measures, deriving new insight
    e.g., \( V_1 = \frac{1}{p_0}, \) LRPT minimizes \( R \)
  - Checking the correctness of measured parameters

Questions?
CS 547: Next Steps

- Read LZGS, Chapter 5 for Wednesday
- Read LZGS, Chapter 6 for next Monday
- Homework #2 due 9/24
- keep up with reading & homeworks