Lecture 9

Prof. Krishna R. Pattipati

Dept. of Electrical and Computer Engineering
University of Connecticut
Contact: krishna@engr.uconn.edu (860) 486-2890

EE 336
Stochastic Models for the Analysis of Computer Systems and Communication Networks

Outline of Lecture 9

- Aggregation and Disaggregation Methods
- Hierarchical Queuing Networks
- Product-form Equivalents of Non-product-form Networks
- M|G|1 Queue
- Application to ARQ Protocol Analysis

Why Aggregation?

- System characteristics that suggest aggregation:

  • Models which represent systems very realistically often *don't have* exact analytical solution (e.g., product form)

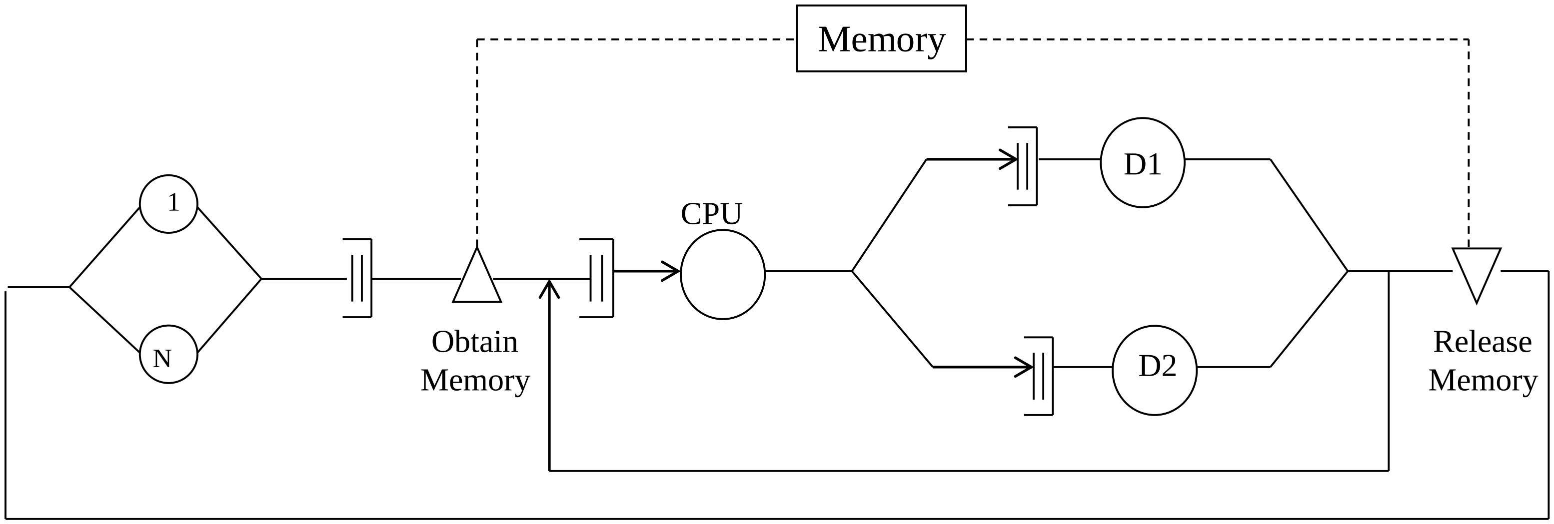
  Q1: Suppose we solve from performance estimates using a less realistic model. Are there any significant differences between the output of realistic model and less realistic model ?

  Q2: What is the error introduced? … active research area.

- Multiple resource holding or *simultaneous resource possession*:

  • A customer (or a job) holds more than one resource at the same time that is, *the customer is in more than one queue at the same time*

Why Aggregation?



- Need memory before being processed by CPU & I/O devices
- CPU & I/O are <u>active</u> resources and memory is a <u>passive</u> resource
- <u>Active</u> resources have service time distribution associated with them, <u>passive</u> resources have no such characteristics.
- Does not satisfy product form
- Can model it as a Markov chain. The number of states explode.
  For example, if N=50, # of memory partitions=12, and four disks, # of states=75,348 $\Rightarrow$ Need to solve $\underline{p} = P^T \underline{p}$ where P is $75{,}348 \times 75{,}348$ matrix

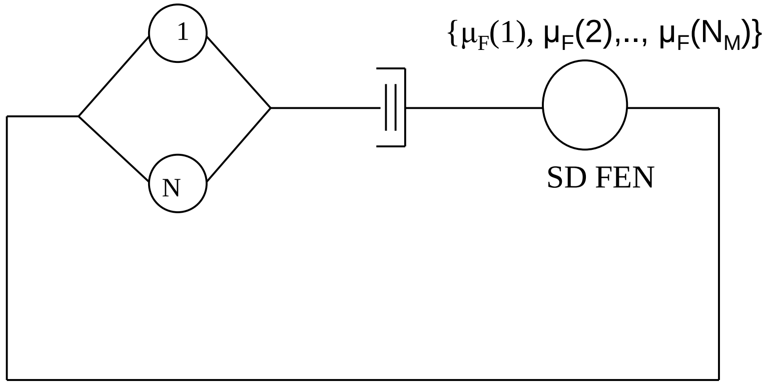
Solution Approach - 1

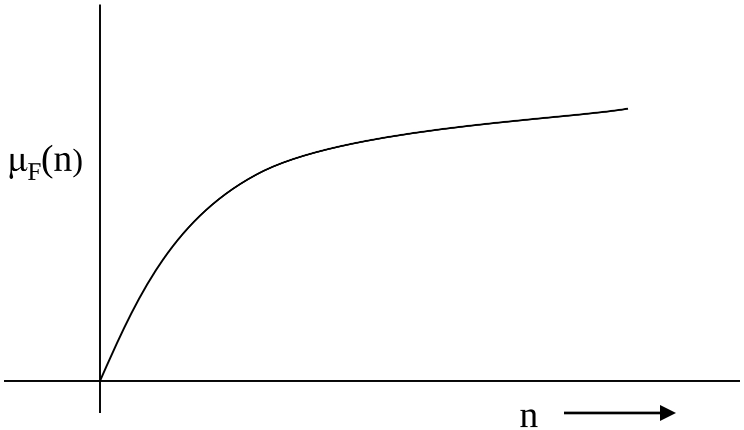
- Solution Approach:

(1) Replace CPU- I/O subsystem by a flow equivalent node (FEN) using MVA. If N_M is the number of memory partitions and each job requires one memory partition, we have:

$$\mu_F(n) = \begin{cases} \mu_F(n), & n=1, 2, \ldots, N_m \\ \mu_F(N_m), & n = N_m + 1, \ldots, N \end{cases}$$

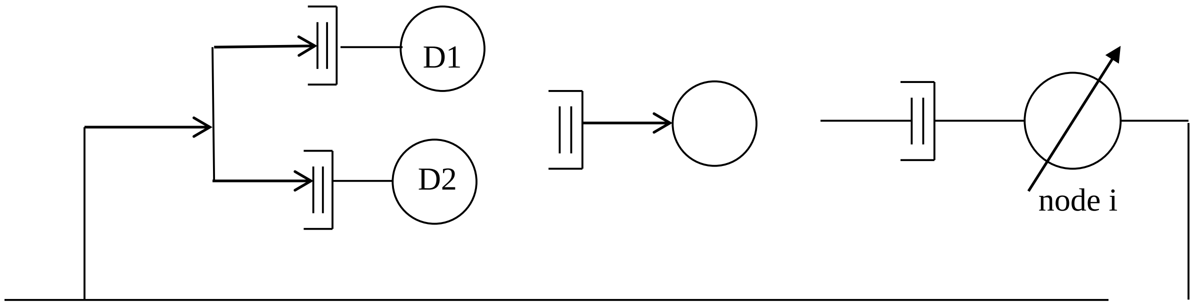
(2) Solve simple network with one infinite server and one state-dependent node





Solution Approach - 2

(3) Disaggregate to obtain measures at CPU and Disks. How? Why does this DECOMPOSITION work? Norton's theorem, Decomposition (aggregation) theorem.



$p_i\,(k|N)$ ~ k at node i and N-k at all node other than i

Know $p_i(k/n) = \dfrac{G_{M-\{i\}}(N-k)\; Y_i(k)}{G_M(N)}$

Since $G_M(N) = \sum_{k=0}^{N} G_{M-\{i\}}(N-k)\; Y_i(k)$

For product-form networks, decomposition is exact. But, also works "good" for non-product-form networks

$G_{M-\{i\}}(N-k)$ can be considered as the capacity function of flow of equivalent node

$Y^i_{FE}(N-k)$ as far as the analysis of subsystem i is concerned

Why $\mu_{FE}(n)$?

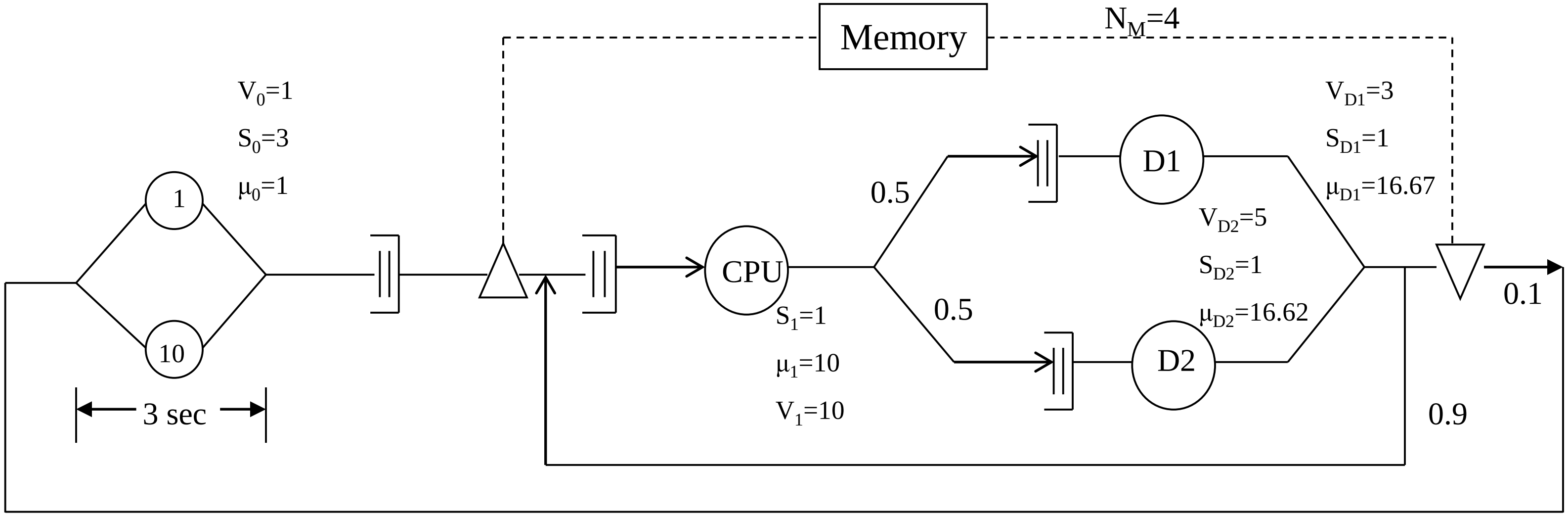
$$Y^i_{FE}(n) = \frac{(v_i s_i)^n}{\prod_{i=1}^{n} \mu_{FE}(l)}$$

$$v_i = 1,\; s_i = 1,\; \mu_{FE}(n) = \frac{Y^i_{FE}(n-1)}{Y^i_{FE}(n)} = \frac{G_{M-\{i\}}(n-1)}{G_{M-\{i\}}(n)} = X^i(n)$$

Throughput of subnetwork with node *i* removed

Illustrative Example 1-1

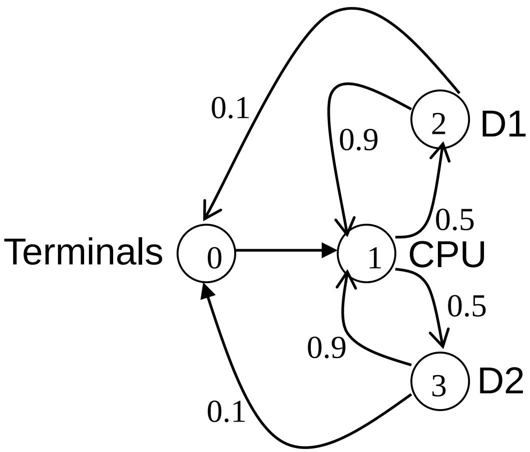
- Simultaneous Resource Possession



$$v_0 = (v_2 + v_3)\,0.1 = 1 \;\Rightarrow\; v_2 + v_3 = 10$$

$$v_1 = 0.9\,(v_2 + v_3) + v_0 \;\Rightarrow\; v_1 = 10$$

$$v_2 = v_3 = 0.5 v_1 \;\Rightarrow\; v_2 = v_3 = 5$$

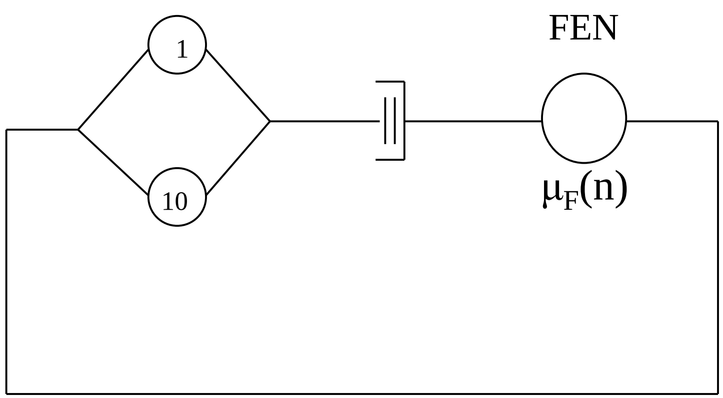


Illustrative Example 1-2

1) Solve CPU-I/O subsystem for populations *n=1, 2*,3, *4*

$$\mu_F(1) = 0.909$$
$$\mu_F(2) = 1.341$$
$$\mu_F(3) = 1.583$$
$$\mu_F(4) = 1.729$$
$$\mu_F(n) = 1.729, \quad n \geq 4$$

2) Solve the smaller network



Throughput X(10) = 1.65
Q_0= 4.23, Q_{FEN} = 5.77,
R_0=3, R_{FEN}=3.58 sec

Illustrative Example 1- 3

- Suppose we want performance measures at CPU and disk also.

  $\Rightarrow$ Need to disaggregate!!!

3) Disaggregation: node $i \in$ subnetwork used to get FEN

$$p_i(k/N) = \sum_{q=k}^{\min(N_m, N)} p_{FE}(q/N) \cdot p_i^{(s)}(k/q) \quad k=0, 1, 2,..., N_m$$

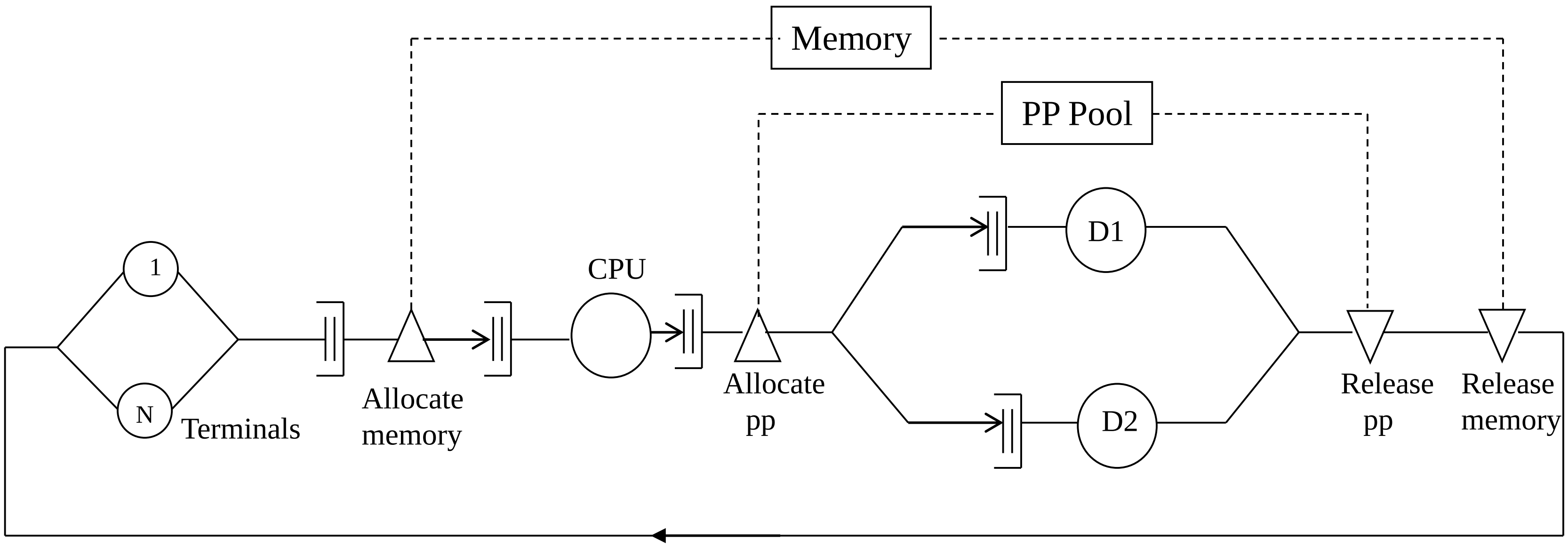
$p_i^{(s)}(k/q) =$ prob. of k customers at node i given q customers in the subnetwork s

$$Q_i(N) = \sum_{k=1}^{\min(N_m, N)} k p_i(k/N)$$

We can extend this idea to any number of subnetworks

Illustrative Example 2- 1

Example: CDC 6600 series computer



- Solve disk subsystem $\mu_{FE}^{D}(1)\ldots\ldots\mu_{FE}^{D}(N_p)$
- Solve CPU-FEN^D subsystem to obtain FEN^C

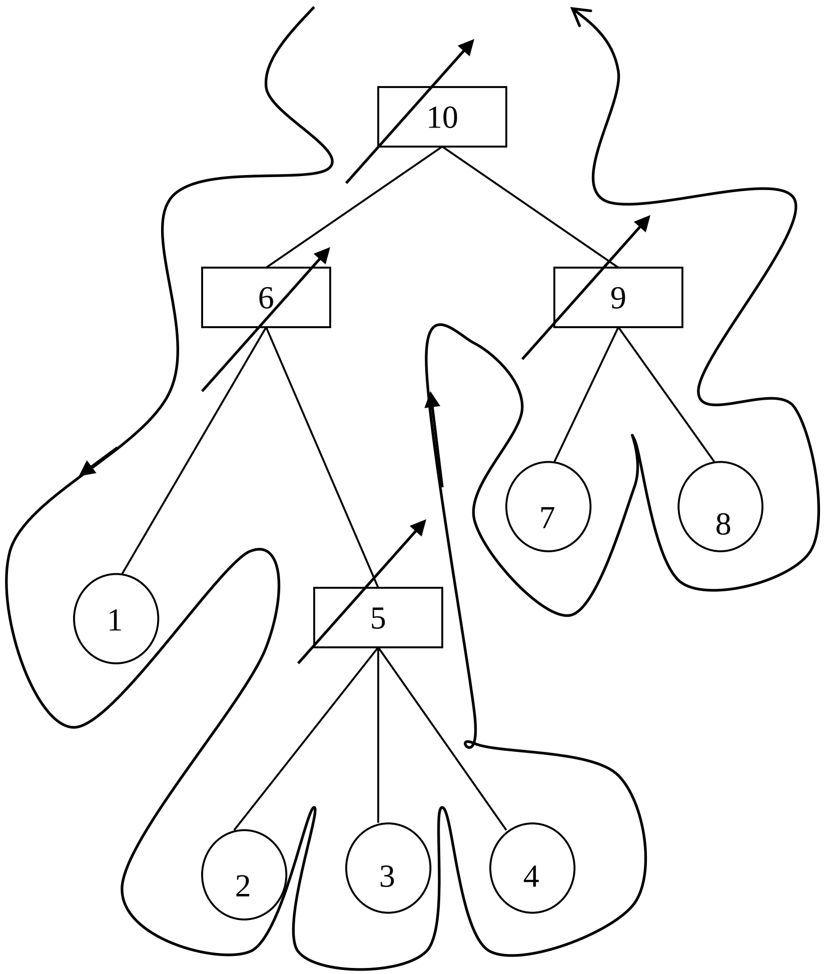
$$\mu_{FE}^{c}(1)\ldots\ldots\mu_{FE}^{c}(N_m)$$

- Solve Terminal-FEN^C subsystem
- Disaggregate hierarchically

PP: Peripheral Processors

Multi-level Networks -1

- Multi-level networks



- Depth first node decomposition
- Aggregate
- Disaggregate

Another reason why decomposition works?

"Interactions within a subnetwork are much more frequent than interactions between subnetworks"…weakly-coupled subnetworks

Example: Transitions between CPU-I/O subsystems are much more frequent than transitions from CPU-I/O to terminals

Multi-level Networks -2

- Open problem:
  - Error involved in aggregation as a function of coupling

    ∃ vast literature on singular perturbation theory in control theory. See. Courtois, *CACM*, 18, 1975, pp. 371-377

    "Decomposability: Queuing and Computer Systems Applications," Academic Press, 1977

- The aggregation technique extends naturally to multi-class networks

$$\mu_{Fj}(\underline{n}) = X_j(\underline{n})$$

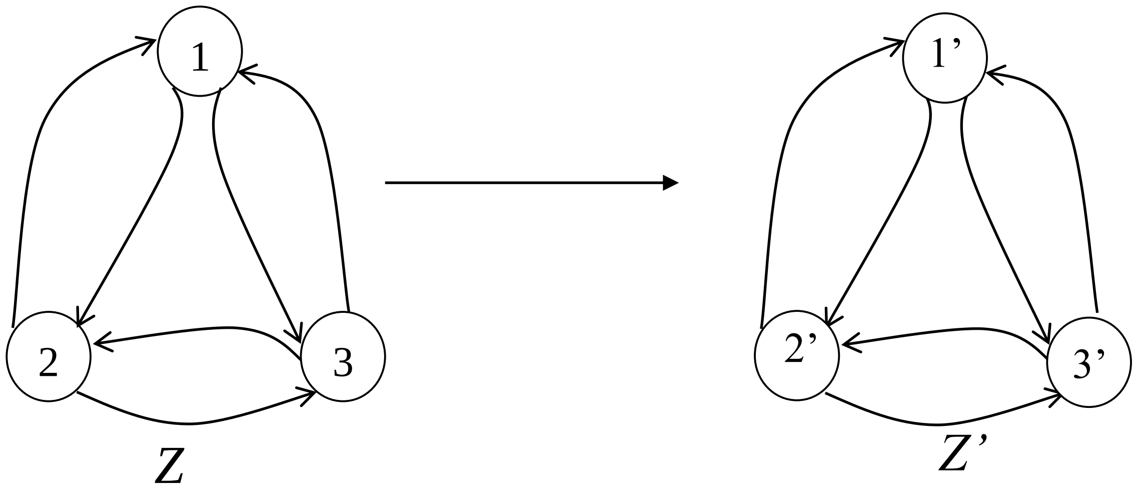
But computational requirements explode!!

∃ Several approximation schemes, however. See References

Product-form Equivalents - 1

- Product-form equivalent of non-product form networks

"Given a general network *Z*. Find an equivalent product-from network *Z'* "



Construct *Z'* ∋ performance measures for each *i'* in *Z'* are *close* to those of the corresponding *i* in *Z*. We call *Z'*, the product-form approximation to *Z*.

Product-form Equivalents - 2

- Premise: 1. A network with two nodes (one general represented by exponentia1 stages i,e., $f(s)=\sum_{i=1}^{L}\alpha_i e^{-\gamma_i s}$) and the other a state-dependent FEN is tractable. Solve via Markov chain techniques. We discussed this in Lecture 5.
  2. If we have *Z*∋ node *1* does not satisfy product-form and node *2* satisfies product from, then we can construct a *Z'*∋ node *1'* in *Z'* behaves like node *1* in *Z* and *1'* satisfies product-form requirements
- Procedure: Want *Z'* from *Z* where only one node (node *i*) does not satisfy product form. Node *i'* ∋ performance statistics of *i* in *Z* = performance statistics of *i'* in *Z'*

1) $Y_{FE}(n) = G_{M-\{i\}}(n)$ and $\mu_{FE}(n) = X^{(i)}(n) = \dfrac{Y_{FE}(n-1)}{Y_{FE}(n)}$

2) solve 2-queue network $(i,\ FE)$ via Markov-chain techniques and get $p_i(n\,|\,N)$

For product from networks, know

$$p_{i'}(n\,|\,N) = \frac{G_{M-\{i\}}(N-n)\ Y_{i'}(n)}{G_M(N)} = \frac{Y_{FE}(N-n)Y_{i'}(n)}{G_M(N)}$$

Product-form Equivalents - 3

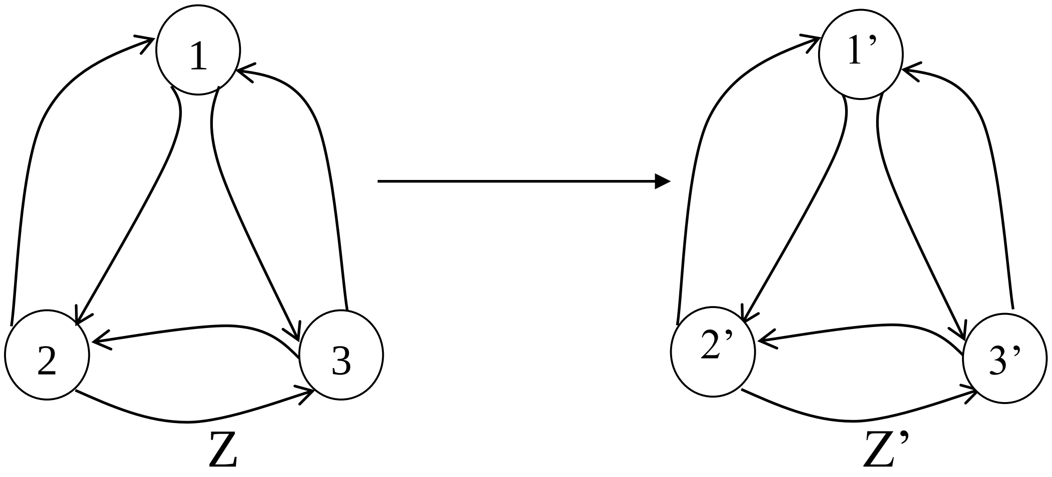
Since

$$\underset{\text{of } Z'}{p_{i'}(n|N)} = \underset{\text{of } Z}{p_i(n|N)}$$

$$Y_{i'}(n) = \frac{G_M(N)\,p_{i'}(n/N)}{Y_{FE}(N-n)}, \quad Y_{i'}(0) = 1$$

So, $$\mu_{i'}(n) = \frac{Y_{i'}(n-1)}{Y_{i'}(n)} = X^{(i)}(N-n+1)\cdot\frac{p_{i'}(n-1|N)}{p_{i'}(n|N)}$$

- So, if we have one non-product from node, the analysis is exact !!!
- What if two or more nodes do not satisfy product-form requirements?



Product-form Equivalents - 4

- Suppose 1 and 2 do not satisfy product-form, but (3) does satisfy. Need an iterative procedure

Step 1: Assume that 1 & 2 satisfy product-form (3 of course does satisfy)

$$1'=1,\ 2'=2,\ 3'=3 \Rightarrow Y_{i'}(n) = Y_i(n)\ \forall i$$

Step 2: Solve *Z'* by any product-form method

Step 3: To Construct a better approximation *1'*

a) *2'* and *3'* are aggregated to a create a FEN

$$\mu_{FE}(n) = X^{(1)}(n)$$

b) Solve $\underbrace{\text{original } 1}$ and *FEN* using Markov chain techniques to obtain $p_1(n/N)$

c) Construct new *1'* that behaves like

$$\boxed{\mu_{i'}(n) = \frac{p_1(n-1/N)}{p_1(n/N)} \cdot X^{(1)}(N-n+1)}$$

Step 4: Construct *2'* new *1'*, *2'* and *3'* from the new *Z'* Solve *Z'* using MVA

Step 5: Compare new *Z'* with the old *Z'*. If close, stop. Otherwise, continue steps 3 and 4

Product-form Equivalents - 5

- General Algorithm:
  1) Start with *Z*, a network with *M* nodes
  2) Assuming *Z'*=*Z*, solve by product-form method
  3) For each i that does not satisfy product-form, do the following
     - Aggregate subnetwork *M-{i}* to get $\mu_{FE}(n)=X^{(i)}(n)$
     - Solve the two-queue network

       (General *i* and *FEN*)
     - Equivalent *i'* will have

$$\mu_{i'}(n) = \frac{p_i(n-1/N)}{p_i(n/N)} \cdot X_{M-\{i\}}(N-n+1)$$

  4) Solve Z' using product-form analysis
  5) Compare *new Z'* statistics with *old Z'* statistics

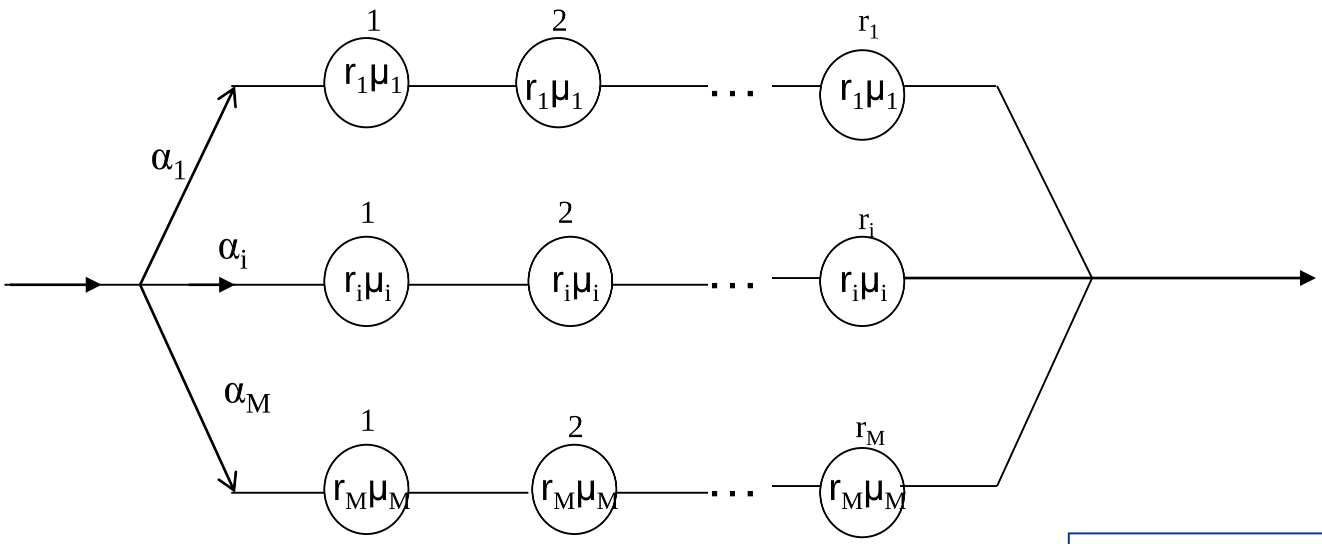
$$\max_i \frac{\left|Q_i^{new} - Q_i^{old}\right|}{N} > \text{ToL}, \text{ go to step 3.}$$

Else Stop

References:
- R.A.Marie, IEEE T - SE, Vol.5, Sept. 1977
- D.Neuse and K.Chandy, Perf.Eval Rev., II, Fall 1982

Solving Two-Node Network -1

- A given service distribution can be approximated arbitrarily closely by a weighted sum of exponential densities ….. Series-parallel stages



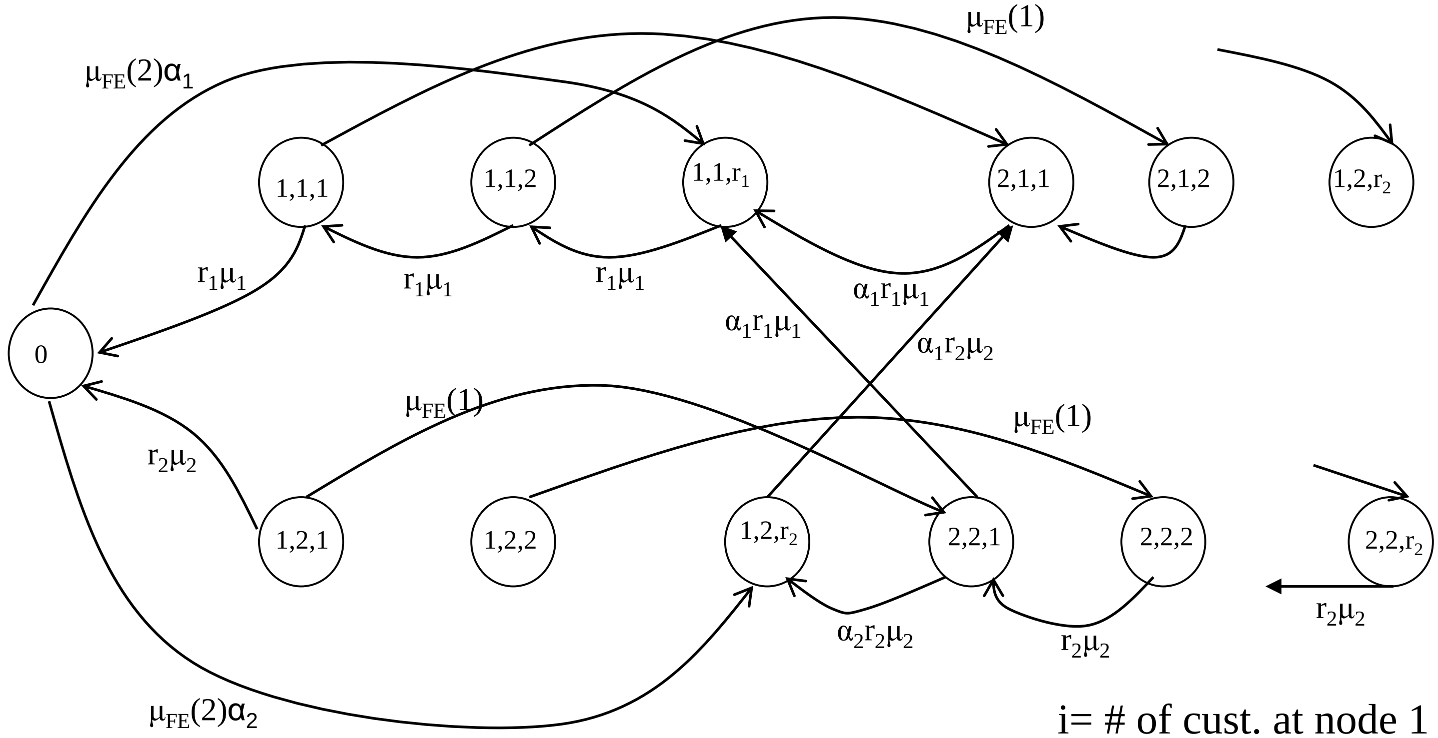
$$f(x) = \sum_{i=1}^{M} \alpha_i \frac{r_i \mu_i (r_i \mu_i)^{r_i - 1} e^{-r_i \mu_i x}}{r_i - 1!}$$

Recall Coxian Representation

Given $f(x)$ or moments of $f(x)$, we can find α_i, r_i and M to match $f(x)$ closely $\Rightarrow$ parameter estimation problem

Solving Two-Node Network -2

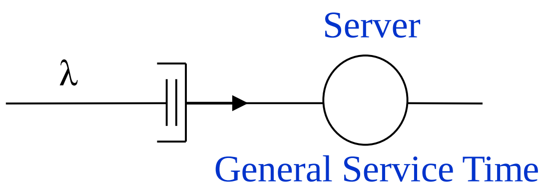
- Suppose we want to solve a two node network where node *1* is represented by series parallel stages with *M=2* and node*2* is a state-dependent node with service rate function $\mu_{FE}(n)$. The population is *N=2*. Then state-transition rate diagram is as follows



Where (i, j, k) $\Rightarrow$ i= # of cust. at node 1
j= parallel state
k= # series states left

M|G|1 Queue - 1

- M/G/1 queue



- • Poisson arrivals, but general service time distribution
- • w/o loss of generality, assume a FCFS service discipline
- • X^i service time of i^{th} arrival
  - *(x^1, x^2, ……)* are *i.i.d.* random variables
  - { x^i }s are independent of inter-arrival times τ_a

- We will show that the waiting time and response times are functions of mean $\bar{X} = \frac{1}{\mu}$ and second moment $E(x^2) = \overline{x^2}$. In particular, we show that

$$\text{Average waiting time } W = \frac{\lambda \overline{X^2}}{2(1-\rho)} = \frac{\rho[1+C_x^2]}{2\mu(1-\rho)}; C_x = \frac{\sigma}{\bar{X}}, \quad \rho = \lambda \bar{X}; \text{ Pallaczek-Khinchn(P-K) formula}$$

$$\text{Average Response time } R = W + \bar{X} = \frac{1}{\mu} + \frac{\rho[1+C_x^2]}{2\mu(1-\rho)}$$

From Little's Formula

$$Q_W = \frac{\rho^2[1+C_x^2]}{2(1-\rho)}; \quad Q = \rho + \frac{\rho^2[1+C_x^2]}{2(1-\rho)}$$

M|G|1 Queue - 2

- Special cases:

$$M/M/1 \;\Rightarrow\; C_x = 1 \;\Rightarrow\; W_E = \frac{\rho}{\mu(1\text{-}\rho)};\; R_E = \frac{1}{\mu(1\text{-}\rho)};\;\; Q_E = \frac{\rho}{1\text{-}\rho}$$

$$M/D/1 \;\Rightarrow\; C_x = 0 \;\Rightarrow\; W_D = \frac{\rho}{2\mu(1\text{-}\rho)};\; R_D = \frac{2-\rho}{2\mu(1\text{-}\rho)} = \frac{(1-\frac{\rho}{2})}{\mu(1\text{-}\rho)};\; Q_D = \frac{\rho(1-\frac{\rho}{2})}{(1\text{-}\rho)} = Q_E - \frac{\rho^2}{2(1\text{-}\rho)}$$

Note: 1) $W_D = W_E/2$

2) $W_D \leq W$

3) ρ small $\Rightarrow Q_E \approx Q_D$; For large ρ $(\rho \approx 1)$, $Q_E = 2Q_D$

- **Will provide an intuitive proof of these results. Rigorous proof in Kleinrock, vol. 1, Ch.5**

Ref.: Bertsekas & Gallagher

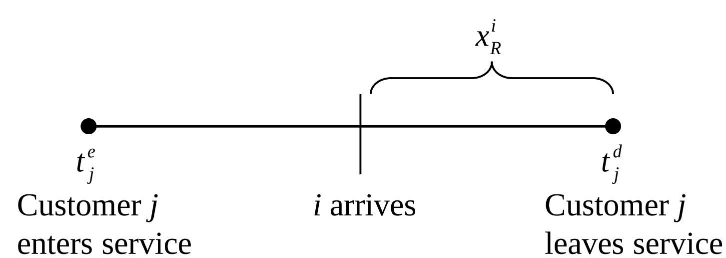
Let W^i = Waiting time in queue of the i^{th} customer

X^i = Service time of i^{th} customer

Q_W^i = number of customer found waiting in queue by the i^{th} customer upon arrival

X_R^i = The residual service time as seen by the i^{th} customer. By this we mean that if customer j is already being served when i arrives X_R^i is the remaining service time until customer j's service time is complete. If no customer is in service, then X_R^i is zero

$$\Rightarrow\; W^i = X_R^i + \sum_{j=i-Q_W^i}^{i-1} X^j$$



M|G|1 Queue - 3

- Taking expectation and noting that X^is are independent by assumption and Q_W^i and X^j are independent $\Rightarrow$

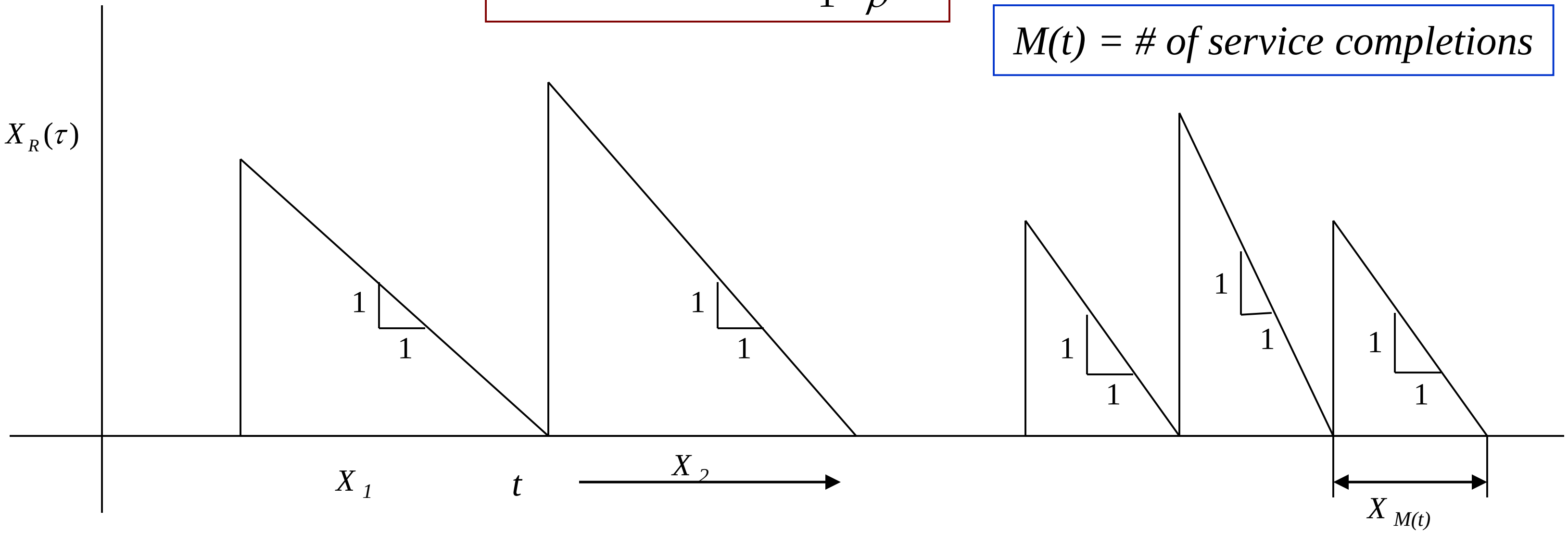
$$E\{W^i\} = E\{X_R^i\} + \bar{X}\, E\{Q_W^i\}$$

Take limit as $i \longrightarrow \infty$

$$W = \overline{X_R} + Q_W \bar{X} \Rightarrow W = \frac{\overline{X_R}}{1-\rho}$$

Note: For M/M/1, $\overline{X_R} = \bar{X}\rho \Rightarrow W = \frac{\rho}{1-\rho} \cdot \bar{X}$

$M(t)$ = # of service completions



M|G|1 Queue - 4

$$\overline{X_R} = \frac{1}{t}\int_0^t X_R(\tau)d\tau = \frac{1}{t}\sum_{i=1}^{M(t)}\frac{X_i^2}{2} = \frac{M(t)}{t}\frac{1}{M(t)}[\sum_{i=1}^{M(t)}\frac{X_i^2}{2}]$$

$$= \frac{1}{2}\lambda\overline{X^2} = \frac{1}{2}\lambda(\overline{X}^2+\sigma^2)$$

$$\therefore\ W = \frac{1}{2}\frac{\lambda\overline{X^2}}{1-\rho} = \frac{\rho(1+C_X^2)}{2(1-\rho)}\overline{X} = \frac{\rho(1+C_X^2)}{2\mu(1-\rho)}$$

Can get this via renewal theory:

$$f_{X_R}(x) = \begin{cases}\delta(x)\ w.p.\ (1-\rho)\\ \dfrac{1-F_X(x)}{2\overline{X}}\ w.p.\ \rho\end{cases}$$

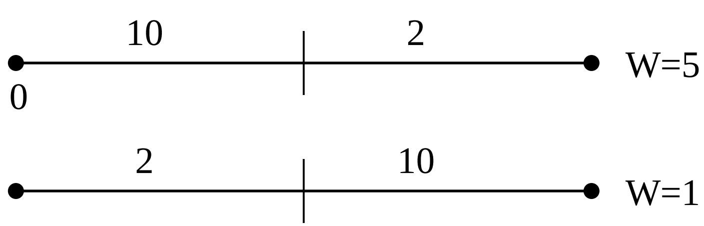
$$\Rightarrow L_{X_R}(s) = (1-\rho) + \rho\frac{1-L_X(s)}{s\overline{X}}$$

$$\Rightarrow \overline{X}_R = -\frac{dL_{X_R}(s)}{ds}\Big|_{s=0} = \rho\frac{\overline{X^2}}{2\overline{X}} = \frac{\lambda\overline{X^2}}{2}$$

- Note: 1) W can be ∞ even if $\rho<1$ e.g., $\frac{\sigma}{\overline{X}}=\infty$

  2) *P-K formula* is valid for any queuing discipline, as long as the order of service is independent of service time

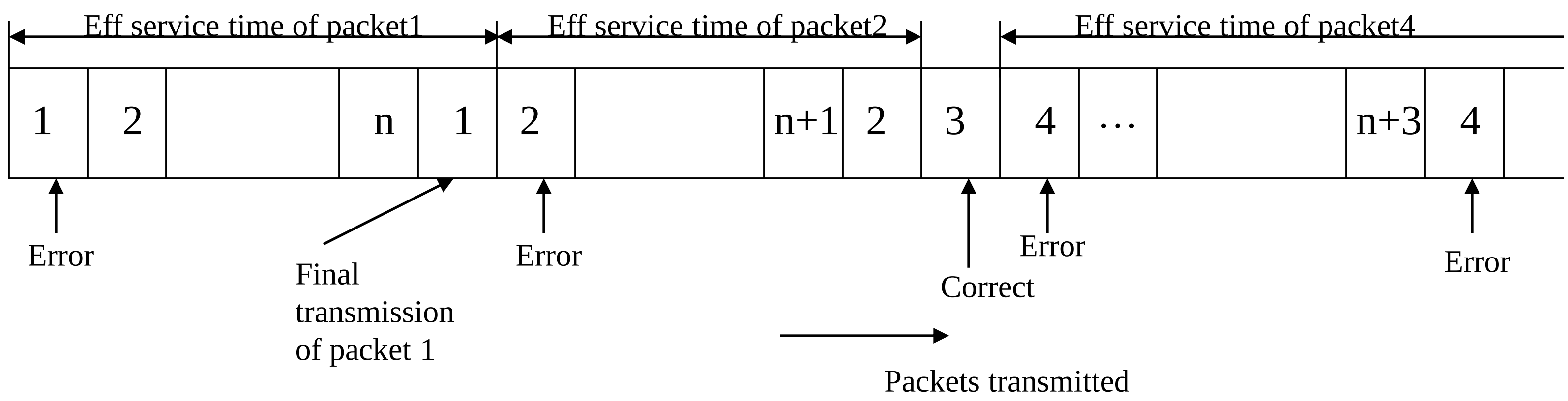
If the service discipline *does* depend on the service time *P-K formula* does not hold!



$\Rightarrow$ W is reduced by serving shorter service time customer

Delay Analysis of ARQ System - 1

- Delay analysis of an Automatic Repeat Request (ARQ) system



- Packets are transmitted in frames that are one time unit long
- There is a maximum wait for an acknowledgement of (n-1) frames before a packet is retransmitted.

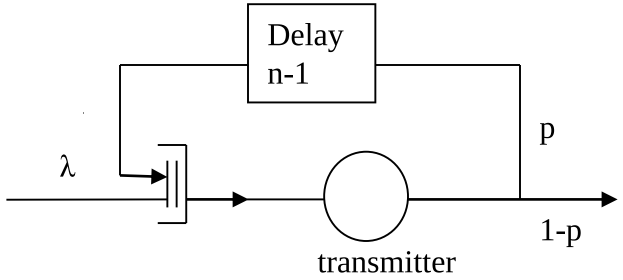
Delay Analysis of ARQ System - 2

- Packet retransmissions are due to:

a) A given packet transmitted in frame *i* may be rejected at the receiver due to errors, in which case the transmitter will transmit packets in frames $(i+1)$ $(i+2)$…$(i+n-1)$ and then go back to retransmit the packet in frame $(i+n)$

b) A packet transmitted in frame i might be accepted at the receiver, but the corresponding acknowledgement may not arrive at the transmitter by the time packet $(i+n-1)$ is completed. This can happen due to errors in the return channel, large propagation delays, etc.

We will assume that retransmissions occur only due to (a). Suppose a packet is rejected at the receiver with probability *p*



Delay Analysis of ARQ System - 3

- Prob { k retransmissions following the last transmission of the previous packet}

$$= \underbrace{(1-p)}_{\text{Success first time}} \quad \underbrace{p^k}_{\text{Occurs after k retransmissions}}$$

- Prob { X= 1+kn } = (1-p) p^k k=0, 1, 2

- Like an M/G/1 queue

$$\bar{X} = \sum_{k=0}^{\infty} (1-p) p^k (1+kn)$$

$$= (1-p)\left[\frac{1}{(1-p)} + \frac{np}{(1-p)^2}\right] = 1 + \frac{np}{(1-p)}$$

$$\overline{X^2} = \sum_{k=0}^{\infty} (1-p) p^k (1+2kn+k^2n^2) = 1 + \frac{2np}{(1-p)} + \frac{n^2(p+p^2)}{(1-p)^2}$$

$$W = \frac{\lambda\left[1 + \frac{2np}{(1-p)} + \frac{n^2(p+p^2)}{(1-p)^2}\right]}{2\left[1-\lambda-\frac{np\lambda}{1-p}\right]} \; ; \; R = W + \bar{X} ; Q = \lambda R$$

As $p \uparrow$ $W \uparrow$ and as $n \uparrow$ $W \uparrow$

in addition, $\lambda < \dfrac{1}{1+\dfrac{np}{1-p}}$ *for stability*

Larger n and larger $p \Rightarrow$ arrival rate should be small

Summary

- Aggregation and Disaggregation Methods
- Hierarchical Queuing Networks
- Product-form Equivalents of Non-product-form Networks
- M|G|1 Queue
- Application to ARQ Protocol Analysis