Computer Communication Networks: Analysis and Design

Professor Yeali S. Sun
Department of Information Management
National Taiwan University
Outline

- Resource sharing 5.1-5.3
- Delay analysis 5.5-5.6
- Capacity assignment 5.7
- Traffic flow 5.8
- Satellite 5.11
- Ground radio 5.12
System design and operation

- The issue of allocating resources among competitive demands
  - e.g., processing capacity, storage capacity and communications capacity.

- Consider a multiaccess system with unpredictable arrival of demands and the demand size

- Goal
  - to find a way to allocate resources to the users’ demands in an effective and efficient way.
Resource Sharing Scenario #1: Uncontrolled Access

- (a) – collection of resources
- (b) – uncontrolled access
Resource Sharing Scenario #1: Uncontrolled Access (cont’d)

(c) – Results of uncontrolled access
- Smashed resource
- Some users do not have any
- Some are happily sharing some resources
Resource Sharing Scenario #2: dedicated resource to each potential user

- Non-work conserving
- Low utilization
Resource Sharing: Law of Large Numbers

- The “collective” demand of “a large population” of random users is very well approximated by the sum of the average demands required by that population.

- The statistical fluctuations in any individual’s demands are smoothed out in the large population case.

- The total demand process appears as a more deterministic (i.e. predictable) demand process.
  - E.g., in life insurance.
Actual Measurement

Poisson Arrival model

Self-similar model

(a) Actual measurement
(b) Synthetic, Poisson model
(c) Synthetic, self-similar model
Resource Sharing Scenario #3: A pool of resources to share

- Private system (e.g., static TDMA) (Figure 5.2)
Resource Sharing Scenario #3: A pool of resources to share (cont’d)

- Each resource is allocated to a user **only when required**.

- A completely shared system with “fany switch” (it can allocate resources when demand)
  - One small resource for each user per time unit (e.g., FDMA) (Figure 5.3)
  - One large shared system (e.g., dynamic TDMA) (Figure 5.4)
Complete Resource Sharing with "Perfect Scheduling": multiple small ones

- One small resource for each user per time unit (e.g., FDMA) (Figure 5.3)
- Possible waste of idle resources due to the division of many small resources
Complete Resource Sharing with “Perfect Scheduling”: large resource shared all

- One large shared system (e.g., dynamic TDMA) (Figure 5.4)
- Shared in a Statistical fashion
- Examples
  - Computer operating system (i.e. fancy switch or scheduler)
    - provide high-performance to a population of users who attempt to share the CPU, memory, second memory devices, printers, network, etc.
  - Client-server system
Basic performance parameters of a resource sharing system

- The **system response time or delay** (e.g., end-to-end network delay)
- The **throughput** (e.g., amount of useful work per time unit)
- The resource **capacity**
- The resource **utilization**
A resource sharing system

- A stream of jobs requests accessing the system resource (arrival process)
- Each requires some number of operations from that resource (service time distribution)
- C – capacity (operations per second)
- Queue – FCFS
- $1/\mu$ – average number of operations required by a job (op/job)
- $\lambda$ – average number of job arrivals per second
- T – response time ($T=W + 1/\mu C$)
- $\rho$ – utilization ($\rho = \lambda / \mu C$)
A resource sharing system (cont’d)

- We want to find the trading relationship among the response time $T$ and the throughput $\lambda$, the resource utilization $\rho$, and the system capacity $C$.

- The system structure has significant effect on this.

- To demonstrate the simplest way of all is often superior to some others.

- Large systems give significantly improved performance as compared to smaller ones.
Possible takings of reducing $T$

- $T = W + 1/ \rho C$
- $\rho = \lambda / \mu C$

**Approach #1:**
- Increase $C$
- Reduce $\rho$

Paying the price of:
- increased system cost
- reducing resource efficiency

**Approach #2 - Not to reduce system efficiency**
- Keep $\rho$ a constant
- Scale up both $\lambda$ and $C$
- Increase system throughput

**Approach #3 - One can maintain a constant $T$ as both $\lambda$ and $\rho$ increase if we permit $C$ to grow less quickly than $\rho$.**
Queueing structures in a resource sharing system (1/2)

Structure (a)
- A collection of $m$ resources
- Each has capacity $C/m$
- Separate queueing - Each is individually accessed by a job stream at rate $\lambda/m$
- $m$ $G/G/1$ systems
- Inefficiency – there may be jobs queued up in front of one of the resources when another one is idle.

Structure (b)
- A single queue accessing the collection of $m$ resources at a total rate $\lambda$
- One $G/G/m$ system
- Both have the same utilization $\rho = \frac{\lambda}{mC}$ (expected fraction of busy resources)
- Jobs are served in FCFS fashion
- Inefficiency – there is no queue but less than all the resources are busy.
Queueing structures in a resource sharing system (1/2)

Structure (c)
- One $G/G/1$ system
- Inefficiency – there may be jobs queued up in front of one of the resources when another one is idle.

Structure (d)
- $m$ times the input rate and $m$ times the resource capacity
- Separate queueing

Structure (e)
- One $G/G/m$ system
- One single queue

Structure (f)
- One big $G/G/1$ system
System Structures

(a) 

(b) 

(c) 

(d) 

(e) 

(f)
Discussion

- All these systems have the same utilization $\rho = \frac{\lambda}{\mu C}$

- Different improvements
Structure (b) - $M/M/m$

- $P_k = \text{prob[system contains } k \text{ jobs]} \quad (k = 0 / \ldots / \infty)$

$$P_k = \begin{cases} 
\frac{p_0(m\rho)^k}{k!} & k \leq m \\
\frac{p_0 \rho^k m^m}{m!} & k \geq m
\end{cases}$$

$$p_0 = \left[ \frac{(m\rho)^m}{(1-\rho)m!} + \sum_{k=0}^{m-1} \frac{(m\rho)^k}{k!} \right]^{-1}$$

- Average response time

$$T = \frac{m}{\mu C} + \frac{P_m}{\mu C(1-\rho)} \quad (5.1)$$

$$T = \frac{1}{\mu(C/m)} + \frac{p_0(m\rho)^m}{\mu C \cdot m!}$$

$$P_m = \frac{p_0(m\rho)^m}{(1-\rho)m!}$$

Average service time
Structure (b) - $M/M/m$ (cont’d)

- $m=1$, minimum $T$
- When keeping $\rho$ constant ($\rho = \frac{\lambda}{\mu C}$) (increasing $\lambda$ and $C$), $T$ is reduced.

- $P_m = \text{prob[the system contains } m \text{ jobs or more]}$, i.e.

$$T = \frac{m}{\mu C} + \frac{P_m}{\mu C(1 - \rho)} \quad (5.1)$$

$$T = \frac{1}{\mu (C/m)} + \frac{P_0 (m\rho)^m}{\mu C \cdot m!}$$
The effect of large systems: normalized average response time (wrt. avg. service time) for structure 5.5(b)

Average service time for a job in one of the m servers

\[
T = \frac{m}{\mu C} \cdot \frac{P_m}{\mu C(1 - \rho)}
\]

(5.1)

\[
\frac{\mu C T}{m} = 1 + \frac{P_m}{m(1 - \rho)}
\]

\[
(\mu C T)/m
\]

- \[\rho = 0, \ P_m = 0 \rightarrow (\mu C T)/m = 1\]
- \[m = 1 \rightarrow \text{approach D/D/1 system (no queue formed until} \ \rho \geq 1\)\]
The effect of large systems: avg. response time with fixed total capacity for structure 5.5(b)

Assumptions
- **constant total capacity**
  \[ C = C_0 = 1 \] and equally shared by \( m \) servers
- \( \rho = \rho_0 = 1 \)
- **Average response time of Structure (c) is superior to that of Structure (b)**
- **Change** \( \rho = \frac{\lambda}{C} \) by changing \( \rho \)

\[
T = \frac{m\rho}{\lambda} + \frac{\rho P_m}{\lambda(1-\rho)} \quad (5.2)
\]

Figure 5.7 Average response time with fixed capacity.
The effect of large systems: avg. response time with fixed arrival rate for structure 5.5(b)

- **Constant**
  \[ \bar{\rho} = \bar{\rho}_0 = 0.8 \]

- **Change**
  \[ \bar{\rho} = \bar{\rho} / \bar{\rho} C, \]
  by changing \[ \bar{\rho} C \]

\[ T = \frac{m \rho}{\lambda} + \frac{\rho P_m}{\lambda (1 - \rho)} \]  
(5.2)

Figure 5.8 Average response time with fixed arrival rate.
Analytical derivations of the relationships $T=(m, \lambda, C)$

- $C$ – total capacity of the $m$ resource system
- $\lambda$ – constant
- The improvement of avg. response time due to single resource systems

$$T(1, \lambda, C) \leq T(m, \lambda, C) \quad m=1,2,3,... \quad (5.3)$$

$T$ depends only on $m$ and $\lambda$ (rather $\lambda$ and $C$)

$$T = \frac{m}{\mu C} + \frac{P_m}{\mu C(1 - \rho)} \quad (5.1)$$

$$T = \frac{1}{\mu (C/m)} + \frac{P_0 (m\rho)^m}{\mu C \cdot m!}$$

$$P_0 = \left[ \frac{(m\rho)^m}{(1 - \rho)m!} + \sum_{k=0}^{m-1} \frac{(m\rho)^k}{k!} \right]^{-1}$$
Analytical derivations of the relationships $T=(m, \lambda, C)$

- **Scaling factor $a$**

$$T = \frac{m}{\mu C} + \frac{P_m}{\mu C(1-\rho)}$$  \hspace{1cm} (5.1)

$$T = \frac{1}{\mu (C/m)} + \frac{P_0(m\rho)^m}{\mu C \cdot m!}$$

$$T = \frac{m}{\mu aC} + \frac{P_m}{\mu aC(1-\rho)}$$

$$T = \frac{1}{a} \left( \frac{m}{\mu C} + \frac{P_m}{\mu C(1-\rho)} \right)$$

$$T(m, a\lambda, aC) = \frac{1}{a} T(m, \lambda, C)$$  \hspace{1cm} (5.4)

- $\lambda = \lambda / aC$
- $\rho = a \rho / aC$
- $\rho$ remains as a constant

With $a$ times increased arrivals, when the system capacity is $a$ times increased, the avg. response time is $a$ times decreased.
Analytical derivations of the relationships \( T=(m, \lambda, C) \) (cont’d)

- True for avg. waiting time

\[
W(m, a\lambda, aC) = \frac{1}{a} W(m, \lambda, C) \tag{5.5}
\]

- With different increase factor \( a \) and \( b \) and \( m=1 \), we have

\[
T(1, a\lambda, bC) = \frac{(1-\rho)}{b[1-\rho(a/b)]} T(1, \lambda, C)
\]

\[
W(1, a\lambda, bC) = \frac{a(1-\rho)}{b^2[1-\rho(a/b)]} W(1, \lambda, C)
\]

\* \( \rho = \frac{\lambda}{\mu C} \), \( a < \frac{C b}{\mu} \)

\* when \( a=b \) ->

EQ (5.4) & EQ(5.5)
Analytical derivations of the relationships $T=(m, \lambda, C)$ (cont’d)

- Using Little’s result and EQ(5.4)

$$\lambda T(m, \lambda, C) = \bar{N}(m, \lambda, C)$$
$$= \lambda aT(m, a\lambda, aC)$$
$$= \bar{N}(m, a\lambda, aC) \quad (5.6)$$

$$\lambda W(m, \lambda, C) = \bar{N}_q(m, \lambda, C)$$
$$= \lambda aW(m, a\lambda, aC)$$
$$= \bar{N}_q(m, a\lambda, aC) \quad (5.7)$$

- The avg. number of jobs in the system or in the queue does not depend upon the scaling parameters, but remains constant for a given $\rho$. 
Recall

\[ T(1, \lambda, C) \leq T(m, \lambda, C) \quad m = 1, 2, 3, \ldots \quad (5.3) \]

Find

\[ W(1, \lambda, C) \geq W(m, \lambda, C) \quad m = 1, 2, 3, \ldots \quad (5.8) \]

- From EQ(5.1), avg. waiting time is
  \[ T = \frac{m}{\mu C} + \frac{P_m}{\mu C(1 - \rho)} \quad (5.1) \]
  \[ W = \frac{P_m}{\mu C(1 - \rho)} \]

- It depends on \( P_m = \text{prob[ the system contains } m \text{ jobs or more]} \), i.e.
  \[ P_m = \sum_{m=k}^{\infty} P_k \]

- We know \( \rho \) is the avg. fraction of busy resources, i.e.
  \[ \rho = \sum_{k=0}^{m-1} \frac{k}{m} P_k + \sum_{k=m}^{\infty} P_k \]
  \( \rho \geq P_m \)

- We have
  \[ \sum_{k=0}^{\infty} \frac{k}{m} P_k = \mu C(1 - \rho) \]
  \[ \sum_{k=m}^{\infty} P_k = \frac{P_m}{\mu C(1 - \rho)} \]

- In M/M/1
  \[ \rho \geq P_m \]
Discussion (1/2)

- In an M/M/m system,
  - the avg. waiting time decreases with m

\[ W(1, \lambda, C) \geq W(m, \lambda, C) \quad m = 1, 2, 3, \ldots \quad (5.8) \]

- the average response time increases with m

\[ T(1, \lambda, C) \leq T(m, \lambda, C) \quad m = 1, 2, 3, \ldots \quad (5.3) \]
Avg. response time and avg. waiting time at constant loads

$M/M/m$

$\rho = 0.96$

$\rho = 0.3$

Difference: avg. service time

$T$ - faster increase

$W$ - Slower decrease
Discussion (2/2)

- If avg. waiting time is the performance measure, then **partitioned systems** are superior such as in general service center (e.g., banks, ticket booth, etc.).

- If avg. response time is the performance measure, the **single resource systems** are preferred such as in packet networks.
The effect of input rate $\lambda$ on avg. response time $T$ in $M/M/1$

$$T = \frac{1/ \mu C}{1 - \rho}$$

$$\rho / \lambda = \frac{\rho / C}{1 - \rho}$$

- For a fixed $\rho$, when increase $\lambda$, one needs to increase $C$ as well -> although with larger input, having powerful $C$ can reduce avg. response time
System scalability vs. system efficiency with constant avg. response time

- M/M/1

\[ T = \frac{1/\mu C}{1 - \rho} \]

\[ = \frac{\rho / \lambda}{1 - \rho} \]

\[ \rho = \frac{\lambda T}{1 + \lambda T} \]
Resource sharing in $G/G/m$ systems (1/3)

- Want to find the tradeoff relationships between avg. response time and input rate, system capacity and number of servers.
Resource sharing in $G/G/m$ systems (2/3)

\[ T \leq \frac{m \rho}{\lambda} + \frac{C_a^2 + (\rho^2 C_p^2 m) + (m - 1) \rho^2}{2\lambda(1 - \rho)} = T_U(m, \lambda, C) \]  

(5.10)

\[ T_U(m, a\lambda, aC) = \frac{1}{a} T_U(m, \lambda, C) \]  

(5.11)

\[ T(1, \lambda, C) \leq T(m, \lambda, C) \]

- Similar conclusions as in $M/M/1$

\[ T \cong \frac{m}{\mu C} + \frac{\lambda [\sigma_a^2 + \sigma_b^2 / m^2]}{2(1 - \rho)} \]

\[ T \cong \frac{m \rho}{\lambda} + \frac{C_a^2 + \rho^2 C_p^2}{2\lambda(1 - \rho)} = T_U. \]

\[ T \cong \frac{m}{\mu C} + \frac{(C_a^2 / \rho) + \rho C_p^2}{2\mu C(1 - \rho)} = T_U. \]
Resource sharing in $G/G/m$ systems (3/3)

\[ C_b^2 = \mu^2C^2\sigma_b^2 = \mu^2\sigma_p^2 = C_p^2 \]

\[ T \leq \frac{1}{\mu C} + \frac{\lambda(\sigma_a^2 + \sigma_b^2)}{2(1 - \rho)} \]

\[ T \leq \frac{\rho}{\lambda} + \frac{C_a^2 + \rho^2C_p^2}{2\lambda(1 - \rho)} \]

\[ T_U = T_U(1, \lambda, C) \]

\[ T_U(1, a\lambda, aC) = \frac{1}{a}T_U(1, \lambda, C) \quad (5.9) \]

\[ T \leq \frac{m}{\mu C} + \frac{\lambda[\sigma_a^2 + (\sigma_b^2 / m) + (m - 1) / \mu^2C^2]}{2(1 - \rho)} \]
Conclusion (1/2)

- For the system M/M/m, large systems (scaling up the input rate and the system capacity) yield improvements in avg. response times that are proportional to the scaling factor.

- For a given scale factor, the single resource system is superior to the multiple (partitioned) resource system in avg. response time (but with increased avg. waiting time).
Conclusion (2/2)

- The concept of large shared single resources is the direction to improve avg. response time of the system.

- Such improvements come about basically due to the law of large numbers in that the relative statistical fluctuations of input demands are smoothed out.

- How about self-similar network traffic?