Total Marks: 100
Due by 23:59 hours on November 18, 2005. Late Submissions will not be accepted.

Instructions and Submission Guidelines:
Submit your work using the “submit” system. Your submission should consist of the following components: a write-up of the experiment results completed using any word processing software of your choice and submitted in PS or PDF format, and your simulation code. This assignment requires you to draw several graphs. Please use an appropriate graphing tool (e.g., gnuplot, Excel, Maple etc.).

Along with your simulation code, include a “ReadMe” file that describes how to compile and run your program. Note that your program should compile and run on one of the departmental Linux servers.

Finally, your teaching assistant might appreciate receiving a hard copy of your write-up. If possible, please hand-in a hard copy of your work during the normal tutorial hours.

Performance Evaluation of a Video-on-Demand System using CSIM
The purpose of this assignment is to study the performance of a Video-on-Demand system using simulation. In this assignment, you will put into practice several concepts introduced in the lectures including statistical models, input modeling, output data analysis, experiment design, and performance evaluation.

A typical VoD server contains several digitally encoded movies. Using a Set-Top-Box connected to a television set or special-purpose software on a desktop computer, a client may browse and select from the movies available at the server. The server operation is simple. The video server receives requests and services them if the required resources (e.g., disk I/O bandwidth, CPU, memory, network I/O bandwidth etc.) are available; otherwise requests are queued. Queued requests are served when the required resources become available. We are interested in the performance of the VoD system, as measured by the following metrics:

- **Average number of server streams**: To support continuous delivery of media data from the server to a client such that uninterrupted playback is achieved, the server must dedicate appropriate amount of system resources for the full duration of the video. For example, disk I/O bandwidth should be allocated such that video file requested by the client is retrieved from the disk subsystem and delivered in a timely fashion. For simplicity, we will collectively refer to the resource requirements for continuous delivery of a single movie as a “server stream”. Implicit in this definition is the assumption that all movies are of the same quality. The average number of server streams is the time-average number of streams required during a simulation run.

- **Average start-up delay**: The number of server streams that can be simultaneously supported is limited by the hardware/software/network
configuration. Thus, when all available server streams are in use, requests are queued. Neglecting network delays, the start-up latency of a request is defined as the amount of time the client waits before it can begin viewing the requested movie.

- **Average client renege ratio**: Clients are often impatient. They may renege (i.e., quit without obtaining service) if their wait time exceeds a “threshold”. The average client renege ratio measure the fraction of clients that leave without obtaining service.

Your simulation program should model the following four components:

1) **The server operation mode**: The number of server streams $S$ simultaneously supported by the video server is limited by hardware/software constraints. In the default configuration, assume $S=100$.

   The VoD system may operate in either the “naïve” mode or the “batching” mode. The operation mode of the server determines how server streams are utilized. In the naïve mode, the video server schedules a new server stream in response to each client request. Requests are served immediately (i.e., with zero start-up delay) if the server has spare capacity; otherwise, the requests are queued and served using a FIFO policy. The batching mode is similar to the naïve scheme, except that queued requests for the same movie are served using a broadcast technique that allows the server to use a single server stream to deliver content to multiple clients. Of course, broadcasting imposes additional network costs (which we will ignore in this assignment). However, it is noteworthy that in many environments such as wireless LANs and cable/satellite television networks, broadcasting is virtually free.

2) **The video files**: Assume that the video server has $N$ video files, all of the same quality. In the default configuration, assume $N=100$. Model the length of each video using a *normal* distribution with a mean of 110 minutes and a standard deviation of 10 minutes, with the minimum and maximum video lengths capped at 90 minutes and 180 minutes, respectively.

   Not all videos are equally popular. You will model the probability of access to the video files using a *Zipf* distribution, with parameter $N$.

3) **The client requests**: The load offered at the server will be characterized by $\lambda$, the average number of client requests per hour. Requests are assigned to a video according to the distribution discussed above.

   You will consider three traffic models for client request generation namely Bernoulli, ON/OFF, and Poisson. These are discussed next:

   - **The Bernoulli traffic model**: To generate traffic according to this model, divide the simulation time into small, equally-sized, time slots of duration $t$. At the beginning of each time slot, conduct a Bernoulli trial, where each trial results in a success with probability $p$, where $0 \leq p \leq 1$. If a Bernoulli trial is successful, a client request is generated, otherwise not. To generate the desired load, you will have to set $p$ appropriately. For example, if $t = 6$ seconds, and the desired input load is $\lambda = 120$ requests/hour, then you will set $p = (120*6)/3600 = 0.2$.

   - **The ON/OFF traffic model**: Again, consider the simulation time to be divided into small, equally-sized, time slots of duration $t$. This model has two states, an “ON” state and an “OFF” state, and is characterized by two parameters $P_{on}$ and $P_{off}$.

In the ON state, a client request is generated at the beginning of each time slot.
Following generation of a client request, the model can transition to the OFF state in the next time slot with probability $p_{onoff}$. In the OFF state, no client request is generated and there is a probability $p_{offon}$ of entering the ON state in the next time slot. The decision to transition from one state to another is made independently. To generate the desired load $\lambda$, the state transition probabilities and the time slot duration should be chosen appropriately. In all your simulations, assume that $p_{offon} = 0.1$.

- The Poisson traffic model: Client requests are generated based on a Poisson process with interarrival time $1/\lambda$.

4) Client behavior: Real world clients are impatient. Model the time duration clients wait before reneging as a uniform distribution in $[W_{\text{min}}, W_{\text{max}}]$. The default values for $W_{\text{min}}$ and $W_{\text{max}}$ are 1 minute and 10 minutes, respectively.

Using your simulation program, conduct the experiments outlined below. Each simulation run should consist of a one hour warm-up period, followed by a six hour interval during which statistics are collected.

- **Experiment 1:** Set $W_{\text{min}} = \infty$ and the server operation mode to “naïve”. Using the Bernoulli traffic model, vary $\lambda$ from 20 to 400. Repeat this experiment, but with server operation mode set to “batching”. Draw a graph with the average request arrival rate on the x-axis and the average client start-up delay on the y-axis with two lines, one for the naïve scheme and the other for batching scheme. Comment on your results.

- **Experiment 2:** Repeat Experiment 1 using the ON/OFF traffic model. For this model, you need to derive a simple mathematical relationship that expresses $p_{onoff}$ in terms of $p_{offon}$, $\lambda$, and $t$. Use this relationship when generating client arrivals according to this model. Compare your results with that for Experiment 1 and comment.

- **Experiment 3:** Repeat Experiment 1 using the Poisson traffic model. Compare your results with that for Experiments 1 and 2 and comment.

- **Experiment 4:** Set the server operation mode to batching. First, using the Bernoulli traffic model, vary $\lambda$ from 20 to 400. Repeat this experiment using the ON/OFF traffic model and the Poisson traffic model. You will plot two graphs. In the first graph, you will summarize the results for client balking. That is, you will graph three lines (one for each traffic model) on a plot that has the request arrival rate on the x-axis and the percentage of clients reneging on the y-axis. In the second graph, you will plot the results for average client start-up delays. Comment on your results.

- **Experiment 5:** Set $S = \infty$, $N = 1$, and server mode set to naïve. Using the Poisson traffic model, vary $\lambda$ from 5 to 100. Repeat this experiment, but with the server operation mode set to “batching”. Draw a graph with the average request arrival rate on the x-axis and the average number of server streams on the y-axis with two lines, one for the naïve scheme and the other for batching scheme. Comment on your results.

***************************************************************************