# CAS CS 655
# Computer Networks

Abraham Matta

Discrete-Event Simulation



## Discrete-Event Models

- Evolution of the system viewed as:
  < s0, (e0, t0), s1, (e1, t1), s2,... >
- Performance measures (output parameters)
  - average number of packets
  - average response time
  - throughput
- Input parameters
  - constants (deterministic)
  - *random variables* (stochastic / probabilistic)



## Average Performance

- Each set of input parameter values gives a *unique* evolution
- **Goal:** obtain performance measures *averaged* over all such evolutions

## Example

- Single server, single queue
- Service time = 2
- Arrivals: 0, 1, 5, 6, 10, 11, ...
- **Define:** state = packets in queue + service
- Events: arrivals and departures
- Show evolution of # packets in system q(t)
- Compute average Q
- Compute average response time T and throughput λ (rate of departures)
- Little's Law: Q = λ * T
  - holds for *any* general system in *steady-state*!

Abraham Matta -CAS CS 655 at CS @ BU

## Flow of Control

SimulationClock = 0;
Initialize system state and performance counters;
Initialize EventQueue;

While "simulation not over" do
  Pick up (e, t) tuple with ***minimum*** t from EventQueue;
  Call Routine (e, t);
    // update system state and performance counters,
    schedule new events and add them to EventQueue
  SimulationClock ← t;

Compute performance measures;

Abraham Matta -CAS CS 655 at CS @ BU

## Event Routines

Routine (Arrival(n), t) =
  Append (n, t) to the tail of Q;
  **Schedule** (Arrival(n+1), AT(n+1));
  If SystSize = 0 then **Schedule** (Departure(n), t + S(n));

  **UpdatePerformanceCounters**;
  SystSize ← SystSize + 1;

Abraham Matta -CAS CS 655 at CS @ BU

## Event Routines (cont.)

```
Routine (Departure, t) =
        // assumes Q is not empty
        u ← Head(Q).ArrivalTime;
        Remove Head(Q);
        If SystSize >= 2 then
                m ← Head(Q).PacketID;
                Schedule (Departure(m), t + S(m));

        UpdatePerformanceCounters;
        NumDepartures ← NumDepartures + 1;
        SystSize ← SystSize – 1;
        AccumResponseTime ← AccumResponseTime + (t – u);
```



## Performance Measures

```
UpdatePerformanceCounters =
        AccumSystSize ← AccumSystSize +
                        (t – SimulationClock) * SystSize;

At end of simulation :
        AverageResponseTime = AccumResponseTime /
                              NumDepartures;
        AverageSystemSize = AccumSystSize / SimulationClock;
```



## Generating Random Numbers

- **Random**: returns a random variable of *uniform* distribution [0, 1]
- Uniform(a, b) =
  return (a + (b – a) * Random);
- Exponential(s) =
  return (-s * ln(Random));

## Computing steady-state averages

- Ignore transient (warm-up) period
  - Beyond a point where average becomes largely unaffected
- Replicate using different random number seeds
- Compute sample mean: $\overline{X} = \frac{1}{n}\sum_{i=1}^{n} x_i$



## Computing confidence intervals

- Compute: $S^2 = \sum_{i=1}^{n} \frac{(x_i - \overline{X})^2}{n(n-1)}$
- True mean is within $t_{n-1,1-\alpha/2} \times S$ of sample mean with $1-\alpha$ % confidence
- For large *n* (>30), replace critical point of t-distribution by $z_{1-\alpha/2}$

## A.3 COMMONLY USED NORMAL QUANTILES

Table A.3 lists commonly used normal quantiles. The confidence levels listed in the first column are for a two-sided confidence intervals. For example, for a two-sided confidence interval at 99%, $\alpha = 0.01$, $\alpha/2 = 0.005$ and $z_{0.995} = 2.576$. For a one-sided confidence interval at 99%, $\alpha = 0.01$, and $z_{1-\alpha} = 2.326$.

**TABLE A.3 Commonly Used Normal Quantiles**

| Confidence Level (%) | $\alpha$ | $\alpha/2$ | $z_{1-\alpha/2}$ |
|---|---|---|---|
| 20 | 0.8 | 0.4 | 0.253 |
| 40 | 0.6 | 0.3 | 0.524 |
| 60 | 0.4 | 0.2 | 0.842 |
| 68.26 | 0.3174 | 0.1587 | 1.000 |
| 80 | 0.2 | 0.1 | 1.282 |
| 90 | 0.1 | 0.05 | 1.645 |
| 95 | 0.05 | 0.025 | 1.960 |
| 95.46 | 0.0454 | 0.0228 | 2.000 |
| 98 | 0.02 | 0.01 | 2.326 |
| 99 | 0.01 | 0.005 | 2.576 |
| 99.74 | 0.0026 | 0.0013 | 3.000 |
| 99.8 | 0.002 | 0.001 | 3.090 |
| 99.9 | 0.001 | 0.0005 | 3.29 |
| 99.98 | 0.0002 | 0.0001 | 3.72 |



## A.4 QUANTILES OF THE *t* DISTRIBUTION

Table A.4 lists $t_{[p;n]}$. For example, the $t_{[0.95;13]}$ required for a two-sided 90% confidence interval of the mean of a sample of 14 observation is 1.771.

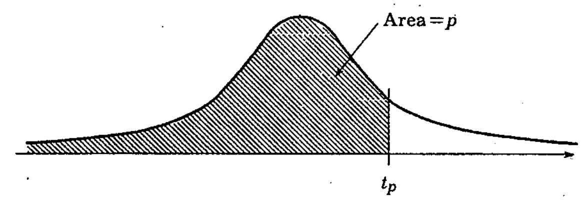


**TABLE A.4 Quantiles of the *t* Distribution**

| $n$ | $p$ = 0.6000 | 0.7000 | 0.8000 | 0.9000 | 0.9500 | 0.9750 | 0.9950 | 0.9995 |
|---|---|---|---|---|---|---|---|---|
| 1 | 0.325 | 0.727 | 1.377 | 3.078 | 6.314 | 12.706 | 63.657 | 636.619 |
| 2 | 0.289 | 0.617 | 1.061 | 1.886 | 2.920 | 4.303 | 9.925 | 31.599 |
| 3 | 0.277 | 0.584 | 0.978 | 1.638 | 2.353 | 3.182 | 5.841 | 12.924 |
| 4 | 0.271 | 0.569 | 0.941 | 1.533 | 2.132 | 2.776 | 4.604 | 8.610 |
| 5 | 0.267 | 0.559 | 0.920 | 1.476 | 2.015 | 2.571 | 4.032 | 6.869 |
| 6 | 0.265 | 0.553 | 0.906 | 1.440 | 1.943 | 2.447 | 3.707 | 5.959 |
| 7 | 0.263 | 0.549 | 0.896 | 1.415 | 1.895 | 2.365 | 3.499 | 5.408 |
| 8 | 0.262 | 0.546 | 0.889 | 1.397 | 1.860 | 2.306 | 3.355 | 5.041 |
| 9 | 0.261 | 0.543 | 0.883 | 1.383 | 1.833 | 2.262 | 3.250 | 4.781 |
| 10 | 0.260 | 0.542 | 0.879 | 1.372 | 1.812 | 2.228 | 3.169 | 4.587 |
| 11 | 0.260 | 0.540 | 0.876 | 1.363 | 1.796 | 2.201 | 3.106 | 4.437 |
| 12 | 0.259 | 0.539 | 0.873 | 1.356 | 1.782 | 2.179 | 3.055 | 4.318 |
| 13 | 0.259 | 0.538 | 0.870 | 1.350 | 1.771 | 2.160 | 3.012 | 4.221 |
| 14 | 0.258 | 0.537 | 0.868 | 1.345 | 1.761 | 2.145 | 2.977 | 4.140 |
| 15 | 0.258 | 0.536 | 0.866 | 1.341 | 1.753 | 2.131 | 2.947 | 4.073 |
| 16 | 0.258 | 0.535 | 0.865 | 1.337 | 1.746 | 2.120 | 2.921 | 4.015 |
| 17 | 0.257 | 0.534 | 0.863 | 1.333 | 1.740 | 2.110 | 2.898 | 3.965 |
| 18 | 0.257 | 0.534 | 0.862 | 1.330 | 1.734 | 2.101 | 2.878 | 3.922 |
| 19 | 0.257 | 0.533 | 0.861 | 1.328 | 1.729 | 2.093 | 2.861 | 3.883 |
| 20 | 0.257 | 0.533 | 0.860 | 1.325 | 1.725 | 2.086 | 2.845 | 3.850 |
| 21 | 0.257 | 0.532 | 0.859 | 1.323 | 1.721 | 2.080 | 2.831 | 3.819 |
| 22 | 0.256 | 0.532 | 0.858 | 1.321 | 1.717 | 2.074 | 2.819 | 3.792 |
| 23 | 0.256 | 0.532 | 0.858 | 1.319 | 1.714 | 2.069 | 2.807 | 3.768 |
| 24 | 0.256 | 0.531 | 0.857 | 1.318 | 1.711 | 2.064 | 2.797 | 3.745 |
| 25 | 0.256 | 0.531 | 0.856 | 1.316 | 1.708 | 2.060 | 2.787 | 3.725 |
| 26 | 0.256 | 0.531 | 0.856 | 1.315 | 1.706 | 2.056 | 2.779 | 3.707 |
| 27 | 0.256 | 0.531 | 0.855 | 1.314 | 1.703 | 2.052 | 2.771 | 3.690 |
| 28 | 0.256 | 0.530 | 0.855 | 1.313 | 1.701 | 2.048 | 2.763 | 3.674 |
| 29 | 0.256 | 0.530 | 0.854 | 1.311 | 1.699 | 2.045 | 2.756 | 3.659 |
| 30 | 0.256 | 0.530 | 0.854 | 1.310 | 1.697 | 2.042 | 2.750 | 3.646 |
| 60 | 0.254 | 0.527 | 0.848 | 1.296 | 1.671 | 2.000 | 2.660 | 3.460 |
| 90 | 0.254 | 0.526 | 0.846 | 1.291 | 1.662 | 1.987 | 2.632 | 3.402 |
| 120 | 0.254 | 0.526 | 0.845 | 1.289 | 1.658 | 1.980 | 2.617 | 3.373 |