Peer to Peer Networks

Lecture notes – 2

**Rehearsal and opening:**

Our goal is to analyze packets in routers. To achieve it we use stochastic models that take as an input ‘arrivals intervals’ and ‘work load’ – each is a process:
A(t) – a process that represents arrivals intervals, described using a distribution function.

Definition: 

B(x) – a process that represents work-loads, also described using a distribution function.

Definition: 
Assumptions:

* The functions A(t) and B(x) hold for all customers/packets – the probabilities are constant and do not depend upon time, type of customer, index etc.
* The different intervals are not correlated. The different workloads are not correlated.

***Part 2 (of course book):***

A queue is characterized using 3 parameters x/y/z:



Server

Queue

Served packet/customer

For example:

M/M/1 describes memorylessness **M**arkovian distribution for both arrivals and workloads, and a single server (1).

M/G/1 describes a **M**arkovian arrival process, **G**eneral workload process and a single server.

This way we can define G/M/1, G/G/1 etc.

During this course we’ll only discuss these cases:

M/M/m – when m is any integer number

M/M/∞ - infinite servers

M/M/1/k - the servers’ queues’ sizes are bounded to k items.

**Service/queue status approach:**

Consider 2 processes: the queue and the service status (assuming one server).

Cn gets into the queue as soon as Cn-1 departs from the queue.

Xn – the time between Cn-1 departure and Cn departure. This is the time the server served Cn request. (if Cn entered service before the departure of Cn-1)

Wn – waiting time – the time between Cn’s arrival to Cn’s queue enter. This is not the total time in the system!

Sn – the total time in the system. it is obvious that Sn = Wn + Xn.

We can examine the convergence of the workload elements. Specifically we look for average Xn, B(x) and b(x) defined as: (density function).



Time

Queue

$$τ\_{n+2}$$

$$τ\_{n}$$

$$τ\_{n+1}$$

Service

Time

Cn

Cn+1++1

Cn+2

Cn

Cn+1

Wn

Xn

Sn

$$t\_{n+2}$$

$$t\_{n+1}$$

Cn

We can also examine the convergence of time elements: 

**Black box queue approach**

Examine the queue from a different perspective – instead of diving into the internal parameters (Wn, Xn) the queue can be described as a black box.

Items

Empty system

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 6 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 5 |  |  |  |  |  |  |  |  |  |  |  |  |  |   |   |   |   |   |  |
| 4 |  |  |  |  |  |  |  |  |  |  |  |  |   |   |   |   |   |   |  |
| 3 |  |  |  |  |  |  |  |   |   |   |   |   |   |   |   |  |  |  |  |
| 2 |  |  |  |  |  |   |   |   |   |   |   |  |  |  |  |  |  |  |  |
| 1 |  |  |   |   |   |   |   |   |   |   |  |  |  |  |  |  |  |  |  |
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |  time |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |  19 |

***General note***: when using the “” notation we’re considering A that has been averaged on time. When specifying we’re considering a temporal average from 0 to t.

**Definitions**:

* .
* , .
* depicted by the upper line in the graph.
**Note**: is not the number of people in the queue- departures process has no affect on.  is a monotonically increasing function.
**equation (1)**: 
* depicted by the lower line in the graph.
* 
**Note:**  is the area encapsulated by the green and red lines ( and )
* . 
**Note:** N(t) demonstrates the black box approach – we look at the number of items in the *system* rather than in the queue. In our case, 
* : Amount of average time spent by customers that arrive before t.
**equation (2):** 
* : average of customers in the system in (0,t)
**equation (3):** 

UPTO HERE 21/3/2012

Conclusion from 3 equations:

 **Little law:**  and in the average case: **\_**

**Notes**:

* Little law assumes **nothing** on the system – neither distributions nor convergence.
* The only limitation is on the last item – we consider the last package before finish serving it but after it got into the queue. This is not a real issue when t goes to infinity. When the system is empty – little law is accurate.

**Back to servers and packets:**

Recall that is the average service time, same as . Note that at any given time, N can be either 1 or 0 and so .  represents the server’s **utilization.** To keep notations consistent we replace  with . By applying little law we get: .

Define: The average number of packets in the **queue** such that: 

Recall that sum of means is the mean of the sum, it is easy to see that: 

server

1 served packet

Server queue

Black box

λ

Define  - meaning that there are i customers in the system at time t - then .

Note that we’re discussing systems where mean of single point in time is identical to the average mean and therefore there’s no need in averaging Pi (though it would be more accurate).

In systems with m servers we define the total utilization as 

This equation describes the average percentage of working servers.

**2.3: Discrete Time Markov Chains:**

A Markov chain is a mathematical tool that helps in describing an element in discrete motion. This topic will be discussed with respect to discrete times and states.

Consider a person moving from town to town. We want to know where this person is at a given point in time.

Abra

Suc.

Zeus

Define a random variable *Xi: person’s position at time i.*

Given that the person ‘jumps’ from point to point randomly, a set of jumps would result a Discrete-time **Markov chain** if the following holds for every n:



The above definition explains why Markovity is used to describe memorylessness, as this is the main feature of these chains: the next step depends solely on the current position.

Define 

Chains that behave according to a constant ‘jumping function’ are called **homogeneous Markov Chains**. At this course we always deal with homogeneous Markov Chains.

Define m-step function:

Recursive calculation of  will result:



(\*) the course book describes more definitions of Markov chains. For example, disconnected chains with 2 broken sets with no edges going between them. we’ll only deal with connected chains.

Say the person was in Abra. We’d like to understand when he’s going to return to Abra – what is the person’s **return-time**.

**Definitions:**

* **Define**: 
* **Define**:  which is the mean of return-time to j.
* If P is the probability to ‘jump’ – we can now define a vector π such that: 🡪 for the transient case
 🡪 for the steady state.
therefore the following holds:

explanation: is the percentage of discrete points in time that j appears at on the time axis. If we take the average time and multiply it by the number of points, we get 1.
if we want to find  based on we can simply calculate:




**Example:**

Abra

Suc.

Zeus

3/4

1/4

1/4

1/4

1/2

1/4

3/4

Based on the example of person travelling in 3 cities we can get 4 equations with 3 variables. Note that since they always be linearly dependent the forth equation is required.



Result for the steady state would be: 

At this case we were only interested finding out what the steady state is, but sometimes the dynamic state is also interesting.

**Dynamic analysis:**

Define a vector of probabilities describing time n: (note that  is a **raw** vector).

We can show that by knowing we could recursively find :

 eventually:



In a similar way we can now define:

 and if it converge - 

**Generating functions with Markov Chains:**

Given we’ll see it’s much easier to calculate using a generating function than simply multiply it by itself multiple times.







So we can take P and create a matching generating function. We can invert the matrix to get. By taking each element and invert it (using methods studied at lecture-1) we ultimately get and can easily calculate.