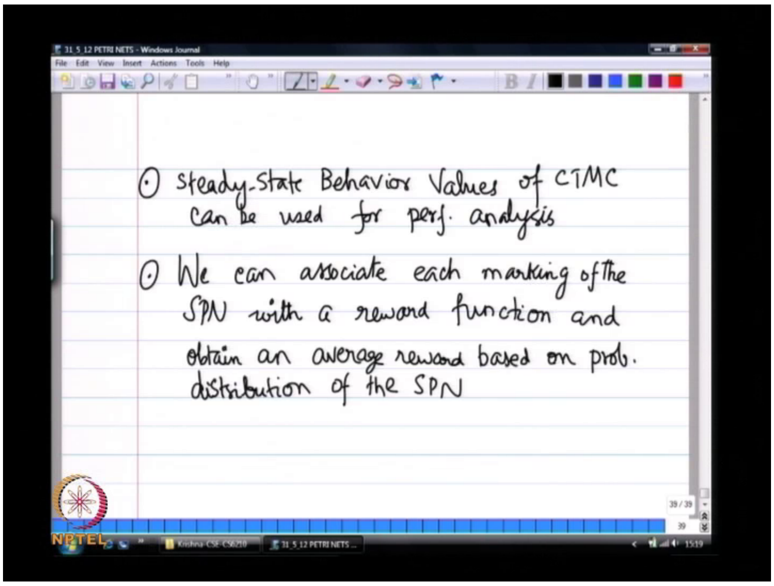
Performance Evaluation of Computer Systems
Prof. Krishna Moorthy Sivalingam
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

Lecture No. # 41
PetriNets - III

Let us great to continue with the petrinets. We look at two more examples and then we will stop that point right. So, now let us write to, so what do we do with this is CTMC, we talked about right taking the Stochastic Petrinets and then generating the corresponding markov chain model from there.

(Refer Slide Time: 00:31)



So, the benefits of that, we can use the steady state right, behavioral remember with we can solve this Marko chain in terms of steady state properties, if it meet some conditions. We have discussed that earlier on right, so we can use the steady state behavioral values of CTMC can be used for performance valuation.

So, they can be used for performance analysis. So, that is the main advantage of looking at this particular class of stochastic petrinet. And in general, what we do is we can associate every marking in the system right, every possible marking basically the state space, that we are referring to each marking of the SPN with some set of reward value. This is what? We used do when we tried find out the expected number of customers in a queue; for example, we said state 0 the reward 0, state 1 reward is 1 in and so on.

Then we tried find they expect number of customer basing sigma reward into corresponding probability. So, like ways we can define some any arbitrary function, when we looked at markov chain for queuing theory that was just the number of the state spaces, each states reward was simply the number of customers buffered in the system at that point in time, but it can be any generic function that you on. So, we can define some of reward function associated that with SPN. And then obtain an averageand average reward based on the probability distribution of the markings. So, this is very generic statement.We would not actually look at an example of this. But in the previous example that we saw right had three states, where is all for p1, p2 and p3 I can simply state that the probability of right reward can be. For example, state the reward of in being in p1, which is both units being operational it is very high.

So, it is like a million dollars for both of them to be operational and if both of them are down and the corresponding reward is like an very small right,0 or something like thatsome negligible amount. And even if one of them is operational it is still right relatively the reward for that is going to be fairly low. So, if you do thatthen you tried calculate the effective reward on the system.

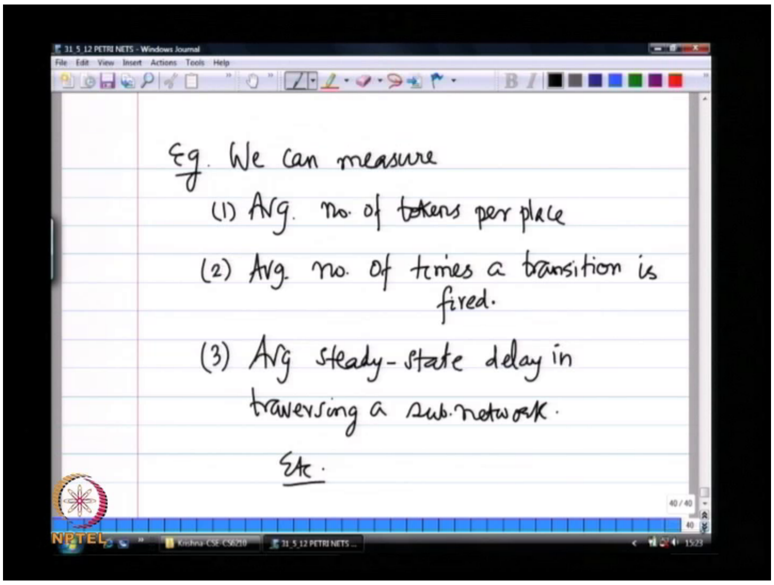
So, by looking at by those looking at p 1 p 2 p 3 multiply by the corresponding rewards will get you the mean reward for the system, and you have to tuned your lambda in mute that is ultimate objective right. If you are going tell that this system will fail in every hundred units of time, that is actually problem not acceptable.

So, will how do see a make it more reliablethis is useful reliable to design to improve the components make show that all the components for better you get reliable components in things like that it. So, the mean time failure should be fairly low in people designeddeals all the time even flight plane for the example you have to make.

So, that collective way the plane right just not you tomiddle of flying you have to make. So, that all the component of such high reliability that over all reliability fairly high. So, this way of for two engine four engine this is a similar thing; if two engines reasons have to at least the one of them is working at least you can fly.

So, that is reward we can associate any reward function you want to with all the states compute the overall system reward. And then used that forevaluationfor further read redesigning the system are for whatever other operation state this performance tuning are for no adding capacity. This system and things like that same thing you can look rewards in the context of buffer for example, if it have less buffer in the system occupied that means, may be less power is consume. So, you want to power consumption low and keep the number of keep the buffer occupancies fairly low.

(Refer Slide Time: 04:28)



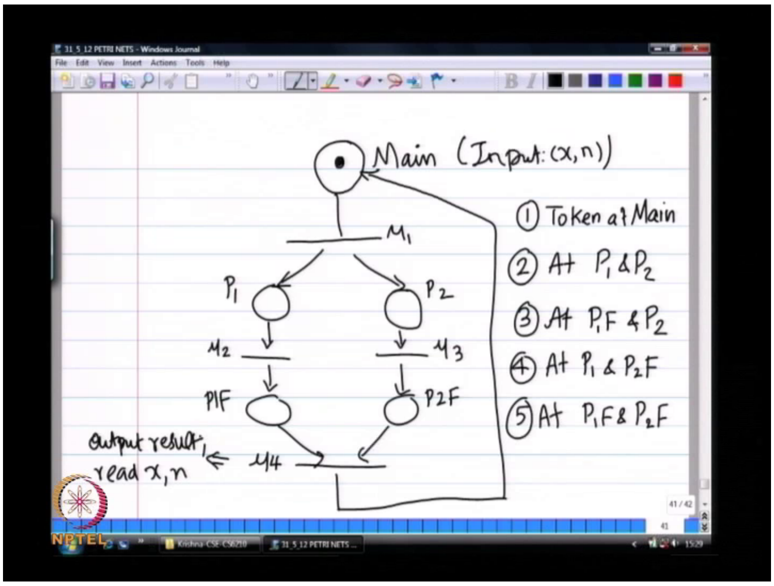
So, that are lots of you know metric setwe can fair you can do for example, you can measure we can measure the average number of token that are there in a given place his can be measured threw. That solve in the CTMC model are you can look at the average number of time set transition is five.

So, how often do we use a particular transitionthat will give the littlebe a corresponding state transition matrix. If you want to from transition is fired, because we want to see when you write we talked about application of Petri net. One of them is also in software design in a system design software design in particular you want to make sure that you have lot ofstatement and the probability of going to particular. If fells classes is very, very low you want to see what is a probability of going actually executingthe that is class that is very low there is no point optimizing the particular piece of code that is another way of (( )).

So,Petri net is also used in the context of the software design for logical system.So, we should look at the and third; of course, we always care about delay. So, we can tried find out the average steady state delay which we know how to do if a find out a number of customers at given point in time in given sub system when we will littleand translate that into the delay also. So, the average steady state delays in traversing a sub-network, so all of these are interesting things to consider etcetera.

So, now, let us look at a simple example, were we can actually little bit mm mostcomputations. So, what we solved early over basic three states system that is go for slightly larger system. So, now this is a program program which we go make it a parallel program are at least two threat program multi threadingand this is at the application level.

(Refer Slide Time: 06:45)



So, let us say that you know that there is this is main right, this place refers to a main program and token being present here indicates the there is some input value. So, whenever get input value is like a running or main program in some set off a while endless for loop. The idea is thatinitial some value for excess in an x are given it talked about next. So, could be two values. So, this represents this token representsx and n two values are rate from the keyboardor default values there in a program already and you want compute whether x of n is bigger n of x is bigger that is the starts to consider.

So, now this is becoming a parallel program right. So, what happens is there is we have a transition which again is assumed to be exponential distribution, some parameter main one.

So, this is some basic some pre process in of to do could be something more if it in this case probably not much pre processing is needed. But, it could be may be database getting some necessary value that your computing system and then you are going to do this processing in parallel, we talked about that little bit earlier to.

So, after processing these to tokens is copy to both this places. So, this is the case of where the data getting copy. So, this is one threat this is and other threat andthis line this transition represents this processing time for this particular threat I let us say this threat take time mu 2 for the rate of processing is mew rate of processing is mu 3 for the threat.

So, then after processing it isyou know, we say that disk the data is completed. So, this may be computingx of n in this computing n of x. So, now the data is available there. So, we will call this state now that is main this is p 1 this is p 2. And this is place can be called p1f and this state can be called p to fand then we have. So, this is a joint operation.

So, come together and that whether transition is and this is computationcomparing these two values doing some processing post getting the data from those two. Then after that this could also include some input processing. So, what we do is this mu 4 could be output result. And it could also have something like read x and n from the keyboard or from some are file something like thatbefore the some amount of time is required and when this is done, we go back to main that is all this is main.

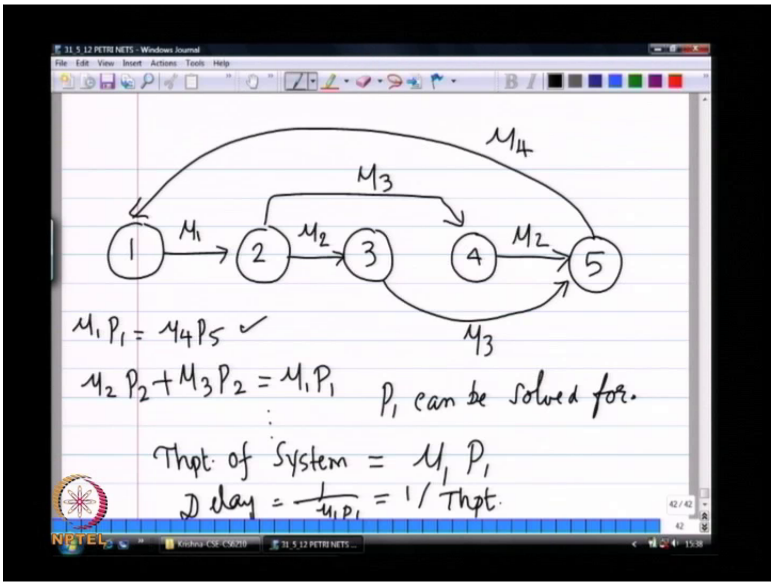
Were any point in time only one data is getting there is one token in this system are we will exceed more than one token, because of application, what this is the behavior of this program. So, the left side computesthis computation right side this, some other computation we join it. And then we send it back no I want to find out how fast can I process numbers in this particular system, where is the bottle neck I want to write see the writhed we can check process data in the system. Thus going to be threw put I want to see the rate of processing of data in this particular system.

So, how do we translate this is SPN this is assumingthat all those times are exponential how shall we translate this SPN into a corresponding CTMC, and then if we can solve for the value that all threw put is what mentions by or we can also compute delay. We can see what is the time average time taken for a packet go through this entire for whilecomputation. So, now, what will be the state space for this particular system whatever all the possible markings,were all if you can findtokensone point their one excess their whatever all the different places of having tokens.

So, will call the state one token is at main. So, when a token is can there numerals in this system now atthen various token b. So, here I am doing application same let us going to p 1 and p 2. So, it is possible that I have tokens at p 1 and p 2 that also possiblebecause that one this mu 2 transition is happening.

So, only happen sometime will mu 2 and mu 3 finishes.Then what are the other combinations at it is possible that p 1 computation is finish. And data setting here cannot be fired until this other part of the computation is finished. So, therefore, here I have a data this is still sitting herebecause this computation is not finished here.Therefore, it is p 1 f and p 2 that is and other marking for the system, and likewise they were p 1 the data sitting here this is finished computation. Therefore it is p 1 and p 2 f any other place; these two p 1 f and p 2 f any other combination, this is only possible combination.

(Refer Slide Time: 12:51)



So, these are the five possible states ofmain mark out change, now let us other state transition diagram for the particular system. So, we call the state, the state one to state two what is the rate of transition I has we need to go back hereright state one to state 2 is mu 1 that is the rate is mu1. So, that is mu 1 then 2 to 3. So, this is when p 1 has finished and sitting p 1 f, but p 2 still processing. Therefore, the rate for transition is simply mu 2 and from here I can also go to the case were, p 2 has finished sitting p 2 f. But, p 1 is still processing and the rate for transition for that is mu 3 then from 4 where can I go to. So, from state four.So, let us look at…

So, there was a packet here and packet here right what will happen I loud only moved this take right were this packet s in both cases like ways in the packet here and packet here when this packet comes here I will be coming over here. So, to go from mm p 1 f and p 2 to p 1 f p 1 of rate is mu 3 right therefore, to go from 3 to 5 the rate is mu 3.

And to go from 4 to 5 where the packet is here token is here and token is here to go this state it is rate mu 2. So, we have mu 2 for this change mu 3 here these are the rate and then when it this finishes I go back to the original state main state with rate mu 5 sorry there was mu 4 mu 4, that is main CTMC. Now, this I can solve I canis any mark out chain solver we can solve this by hand it isfairly straight forward do that.

So, what is the state balance equation again right it is going to be simply mu 1 p 1equals. So, this is a ferment p 1 rate of departure is mu 1 p 1, what is the rate of coming back mu 4 p 5. And then mu 2 p 2 it is assignment mu 2 rate of departure is mu 2 p 2, and the rate of arrival is mu 1 into p 1 no sorry here sorry that is not correct. If I look at this one, if I look atp 2 I can leave with write mu 2 as well as that is also mu 3 p 2, these are the two departuresarriving is mu 1 p 1 and so forth.

We can prettymuch solve for all of this and then now I am interested threw put of the system I can just write down the equation for this actually, we can do not needed the solver for the manualwrite down the equate all the p 1 values. And you will find that you know p 1 can be solved for. In once is solved for this threw put of the system is given by what, how will defined threw put of the system. Threw put of the system defined when whenever main is able to process a new data.

So, when will what is the rate ofnew datagetting process by main, whenever your passing data to the p 2 p 3. So, probability of being in state p 1 which means your enter main and the rate of leavingamount of processing with p 1 has to do. So, that will be given by mu 1 into p 1 this will be the threw put of the system. So, mu 1 equals 5 and p 1 is 0.1 then it will be 0.5 is the effective threw put of the particular system. So, mu 1, no comments mu 4 into mu 4 into p 1 no p 1 is a probability of being that state so the...

So, itwill be also mu 4 into p 5 s by this because it does not matter because that is our mu 1 p 1 of mu 4 p 5 will be the same , what that is our parents equation. So, one of the other you can solve for anywhere p 1 p 2 you will basically solve for all the probabilities and then new essentially take mu 1 p 1 or mu 4 p 5 whatever thethat is the rate of which you leave the that state to go to the next state. That means, that the next processing is happening, that is way will defined threw put mu 4 into mu 4 wise mu 4 mu 4 into p 5 is the probability of going notp 1 say is p 5 is here.

So, if you are in state 5, which is when you have you obtained all this results for those two p 1 of p 2 f, you are now finished your processing with rate mu 4 you are going back to state one the main state right that can also be considered as returning back to may. But by our balance equation it is same is mu one p 1. So, if I am currently in state p 5.

And I finished my processingall there getting the next data,I want to find out when the next data can be processed. So, this is that will be either mu 4 and p 5 or mu 1 into p 1. So, now, if you want to find out the delay on the system, how will you do that we want to find out the average time taken by a packet to go through this, what will be the how will you compute the delay value?

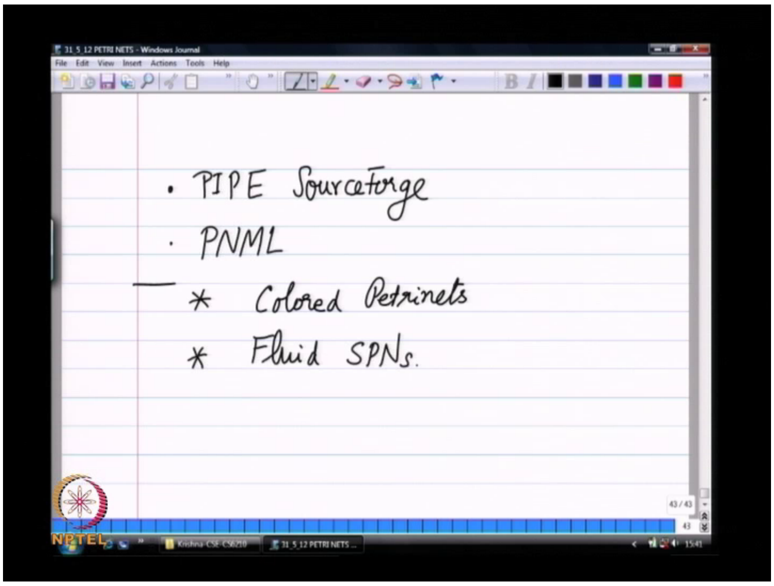
1 by 1 by mu over one over through put will be the total delay in the system one over through put one over mu 1 p 1 will there correct, what is expected number of job in the systemexpected number of job in the system, is exactly oneright that is only one job in the system that is circulating right.

So, if I use little's law can I use little laws here will you get that still one over mu 1 p 1 you will get that, is you are finishing you through put is basically whatever 10 jobs per second. That means, an average will be requiring 0.1 unit of time per job, delay will simply one over mu 1 into p 1 it will be little's law, what will I do well fired find out the what is a what is a arrival write into a system according to little's law. It will be e of n which is one divided by the effective arrival write into the system.

And what is effective arrival write mu 4 into p 5 which is also mu 1 into p 1 for it is for delay equals one over mu more right. So, delayis going to be one over mu 1 into p 1.

So that we can figure by we just solving for this particular equations, so questions on this and you want homework on more complicated not modeling problems. Homework no homeworkpresume that you do not want any homework.

(Refer Slide Time: 21:54)



So, we predict much winding down. So, that is several tools available atnot just at this dimension pipe. So, pipe is a tool we you can actually inputwith graphically interface. So, this is available source forge, I am not sure what text in it is professional extend it is professional well maintained for it is out there, if you look at this says this is a bunch of MSC projects from imperious college in London.

So, about four, five years of m tech equal an project have been then in try to make this we can also do that here if near to. So, that is one particular system that is there I am not it tried the other ones, but there is look at. In fact, that is something calledPetri net. Markup language this also there markup language is also be defined.

But, if you will look at the Wikipedia page it is same is to be default I am not sure, what is the states ofthat is because the people wanted to havesimple way to write. This specifications for a Petri net in the give that any simulator any tool I will take their; otherwise what happened every tools own way of representing the data and putting data, and things like that if there is a common markup language. Then you can always ported between different system the ultimately, what we have a just want a specified set of art and a transition and the places in a particular system.

So, there is something also that you can look at. So, besides this people of been fairly active relatively active community it is out there. So, there is extension called colored Petrinets the default Petrinets is seen is all tokens of the same. So, colored petrinets simply mean it different data types. So, different tokens will different types of token exist where each token corresponds ofdifferent data type and then you can solve for those.

So, colored Petrinets there is a book lots of books have are available thenother things. So, even the GSP talked about can be even for the extend at with non- exponential delays also, we talked about exponential delay or immediate triggering. But there are possibilities of other types of delay distribution are also possible. And then we have something call fluid SPNs which have not a looked into right, if you look at the way Wikipedia page for (( )) in general the lots of exponential out there.

So, in summery this seen is to be useful tool sometimes it is easy to represents the system in almost looks like a flow chart, it is like a flow chart representation. Only thing it is presides and semantic are well defined. So, you can take that you can feed it and then it we can potentially also simulated it you have to with those specification. So, so useful tool, but unfortunately there is no time tool actually try it out in the context of this course at least, but we can probably add in exercise for somebody try it out the future.