POLITECNICO DI TORINO

# MANAGEMENT AND CONTENT DELIVERY FOR SMART NETWORKS

## Algorithms and Modelling

**A.A. 2015/2016**

**LABORATORY 2**

**DESIGN OF RATE ADAPTIVE ROUTERS**

**BARUSSO FEDERICO**

**CERUTTI MARCO**

**ORLANDO MATTEO**

The goal of the laboratory is to implement a simulator that behaves as a router with rate adaptation, and analyze its performance about packet processing and energy consumption.

The focus is only on the output port of the router, and we make some assumptions

- The router can work at two different rates: *high* and *low* with capacity $C_H$ and $C_L$ that consumes respectively $P_H$ and $P_L$
- The time to switch from the *high* state to the *low* is $T_{HL}$ and the time to go from *low* to *high* is $T_{LH}$
- The routers know at each time the traffic load
- The input traffic is generated as a Markov Modulated Poisson Process with two states: *high* and *low*, with arrival rates $\lambda_H$ and $\lambda_H$

The starting point is a simple M/M/1 queue simulator that is customized according the laboratory goals.

Four kind of events are used to design the simulator. Two are the ones common to every router, *arrival* and *departure*, the others two are *change* and *adapt*.

*Arrival* is the event that occurs when a customer arrives, while *departure* is the event that occurs when one of them leaves the server.

*Change* is the event that occurs when the system passes from the *low* state to the *high* one and vice versa, while *adapt* occurs $T_{HL}$ or $T_{LH}$ seconds after *change*, to update the parameter to the current status, respectively from *high* to *low* and from *low* to *high*.

The attention is focused on how the performance of the router varies by changing some of the parameter explained before.

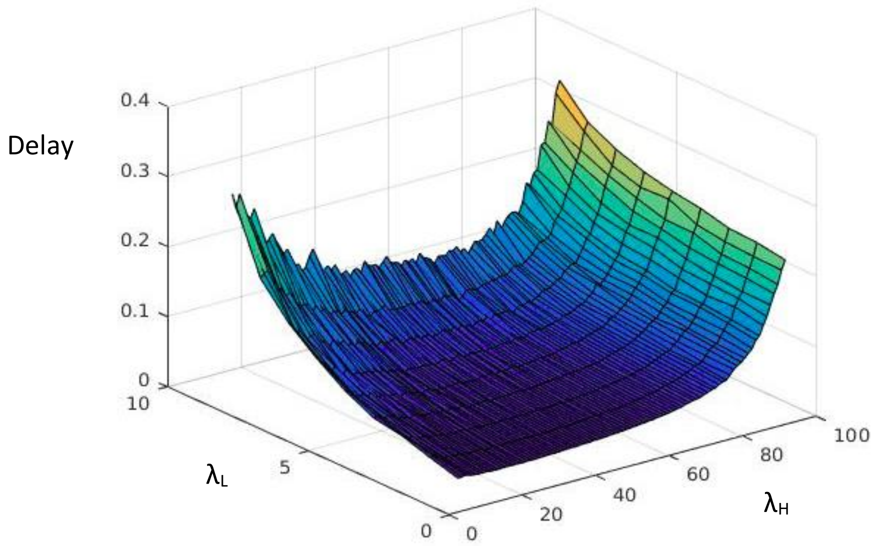
In the first analysis $C_H$ and $C_L$ are kept constant (100 and 10) and also the two power value $P_H$ and $P_L$ (100 and 1) and a series of simulations are run checking all the possible configuration of $\lambda_L$ that goes from 1 to 9 and that $\lambda_H$ goes from 10 to 99.

**Standard Case**

In this case the average time spent in the two state is the same.

| $T_{HL}$ | $T_{LH}$ |
|---|---|
| 0.1 | 0.1 |

In the following picture are plotted the results for the delay.



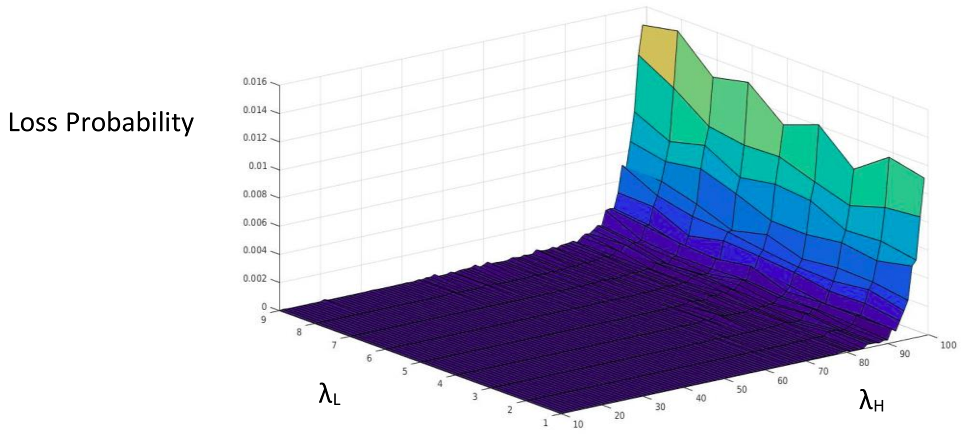

As we can see the average delay increases in two main cases:

- $\lambda_L$ high and $\lambda_H$ high
- $\lambda_L$ high and $\lambda_H$ low

Why this happens in the first case is straightforward: when both the arrival rates are close to the capacity of the server, the load is big and customers may spend a lot of time in the queue before being served.

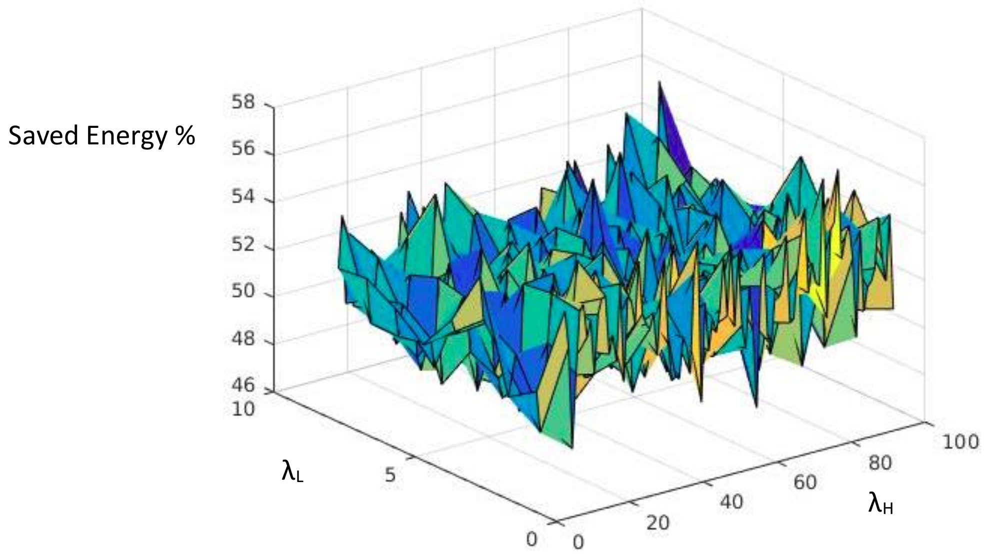
The reason why the delay is high also in the second case is more tricky. As we can see if $\lambda_L$ remains constant and $\lambda_H$ increases, the average delay decrease. This behavior is due to the fact that if $\lambda_H$ increases, the time spent by the system in the high state also increases, and during the high state a lot of customers are served very quickly. That's bring a reduction of the average delay. Instead when $\lambda_L$ is high and $\lambda_H$ is low the system spent about the same time in each state, due to the fact that high values of $\lambda_L$ are very similar to low values of $\lambda_H$, and the two states have the same contribution to value of the average delay. In this case quite the half of the customer is served quickly while the other half is served slowly, so the average delay increases.

In the following picture are plotted the results for the loss probability.




In this case there are no particular behavior: the loss probability increases with the growth of the server's load, as expected.

The percentage of saved energy is plotted in the following figure.



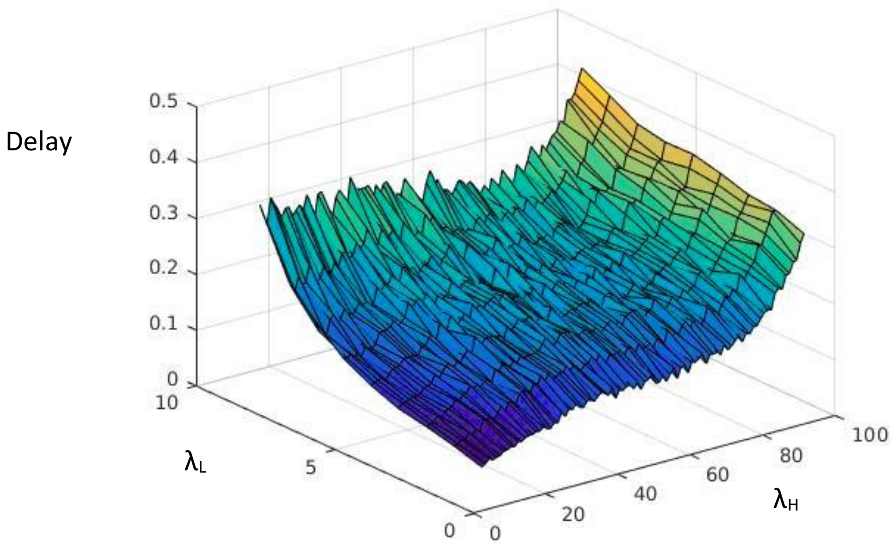

Is possible to see that the energy saving doesn't depend at all on the arrival rates and in average is about 51%.

**Switching Time Variation**

In this simulations the time needed to switch from one state to the other is the only parameter changed. This is done in order to discover how the switching time can bring some differences respect to the previous case, in which switching time was so little that it had no effects.

| $T_{HL}$ | $T_{LH}$ |
|---|---|
| 2.0 | 2.0 |

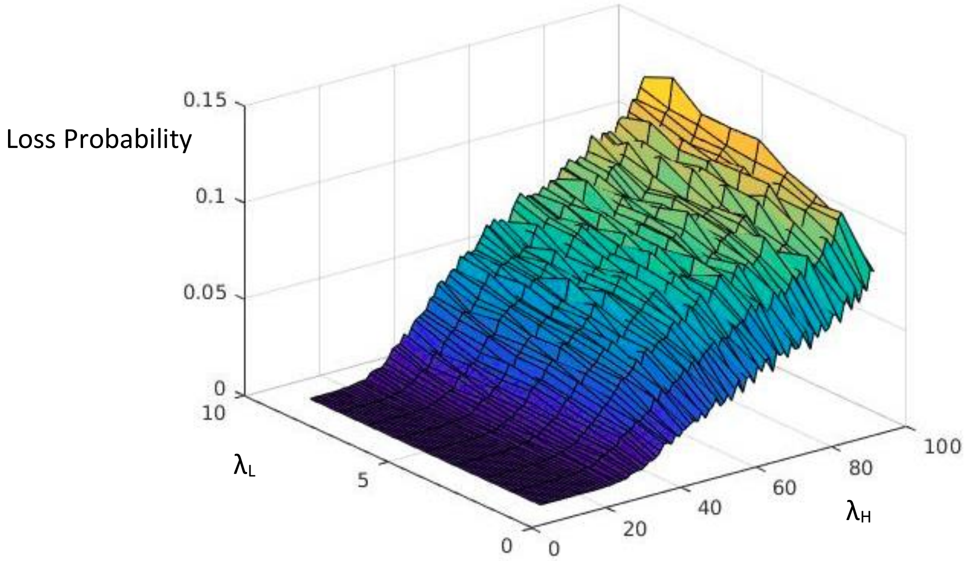
The following pictures represents the plot of the average delay



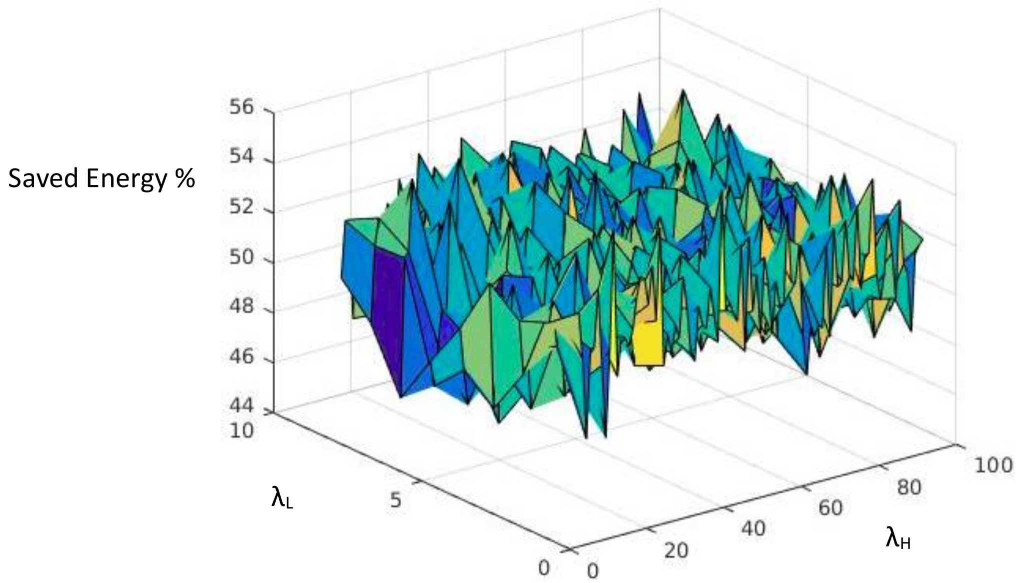

It seems that the trend of this case is similar to the fast-switch case, but smoother and with less difference of values between the high load configurations and the low ones. Is possible to see that there is no more that great improvement in the delay when $\lambda_L$ is high and $\lambda_H$ have a middle value.

This is because the switching time introduces an additional delay in serving customers when the system passes from one state to the other.

The result for the loss probability are represented in the following plot.



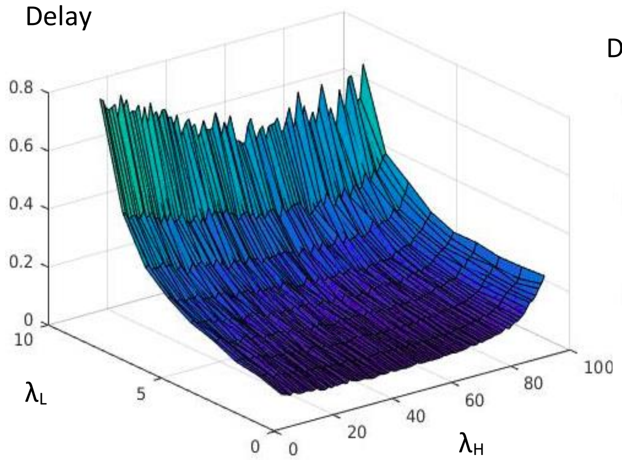

This result is quite different from the fast-switch case. As expected the loss probability increases with the load of the server, but in this situation it starts to increase much earlier and reaches ten times the value of the other case at the worst configuration.

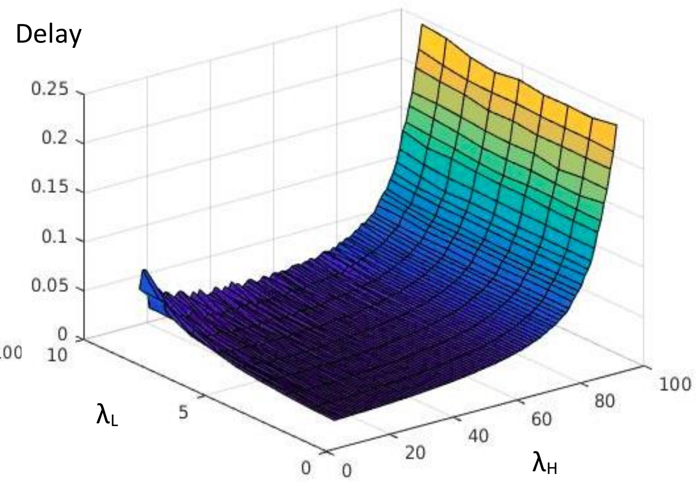



For the saved energy instead the results are really similar to the previous one but with a slightly difference in the mean value 50%. Having a so little difference is possible to say that the two cases are not influenced by the switching time.

**Variation of state occupancy probability**

With this simulation, the goal was to understand how the performance is influenced by the average time spent in one of the state respect to the other. To make this, two simulations were run, where the time spent in the two states was in average one ten times the other.

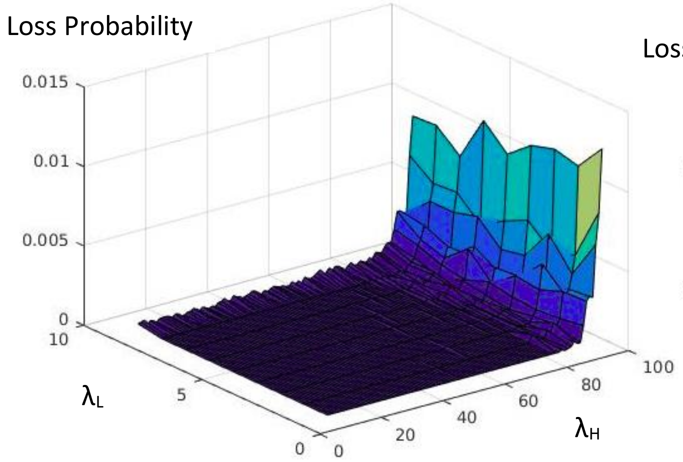


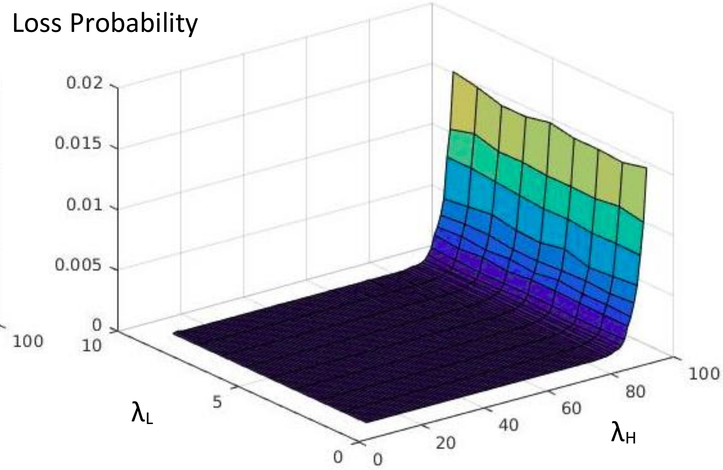




The first plot corresponds to the configuration with the most of the time spent in the *high* state, while the second one is obtained by the configuration with most of the time spent in the *low* state.

In the first configuration the delay depends quite only on the variation of the $\lambda_L$ while in the second one depends on the variation of $\lambda_H$. We can also see that the average delay is lower when the router spent most of the time in the *low* state rather than in the *high* one.

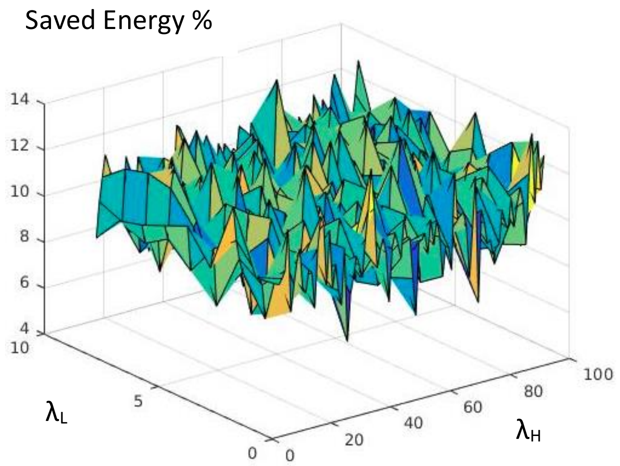
The loss probability instead seems to depend only on the increasing of the $\lambda_H$.

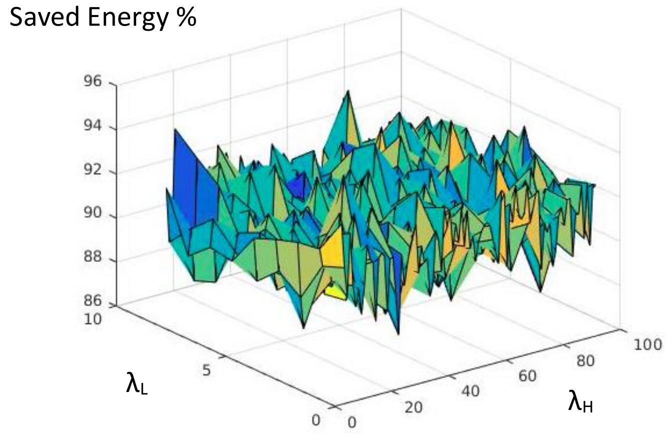


As we expect from the previous simulation the trend of the saved energy does not depend on the variation of the arrival rates, but in this case is possible to see that there is a huge difference in the amount of saved energy.

In fact, is possible to see that when the system passes most of the time in the *low* state the amount of saved energy is huge, that is exactly the reason why double-state routers are used.




In conclusion is possible to tell that while in case a router needs to stay most of the time in a configuration similar to the *high* state, double-state router does not give a great improvement, also if only a 10% of saved energy can be important. But if the router can serve a good percentage of its customer with lower rate, double-state router can be very useful, because the bigger is the percentage of "low rate customer" it needs to serve, the higher is the amount of saved-energy.