Crisp values (file:RESULTS-ZNN.xlsx)

The table below (and its companion chart) show the percentage of requests which were **denied** because there was no server available to accept them at that time, and the percentage of requests that were **overdue** because they were initially accepted by a server but they were finally responded later than the required threshold (due to crisp comparisons).

There are two sources of uncertainty that may cause a request is overdue:

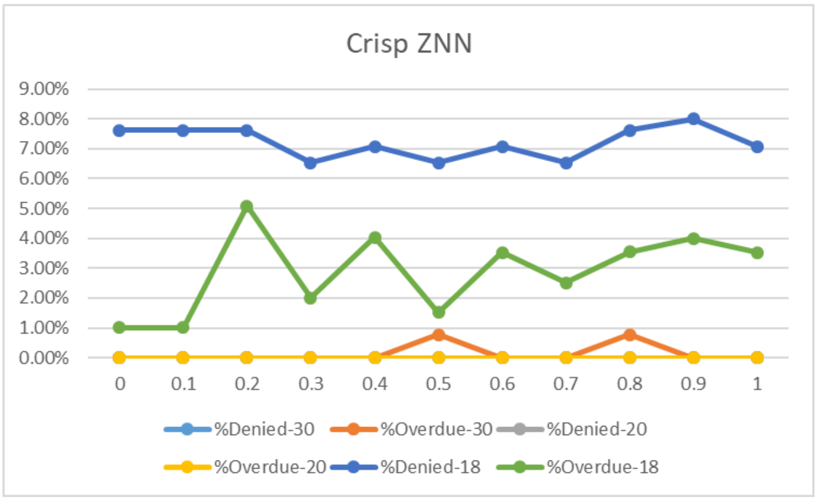
- The processing time of a request is not a fixed value but a random variable. This can cause the actual processing time to be longer than the estimated processing time. Thus, a server may accept a request because according to its calculations it is able to respond to it in the required time, but then the actual processing time is longer than expected and the response is delayed.
- Clocks have some imprecision. We have assumed that deviations are only of micro-time units (i.e. 10E-6) but this can cause some comparisons between time units to fail. This can also cause that some requests end up being delayed, because when the system checks if a request has finished (timeNow>=request.finishTime) the comparison can fail and the request has to wait for the next cycle to be responded.

This version assumes that all values are crisp, and no treatment of uncertainty is carried out. The tests are performed assuming the following parameters:

- Number of clients issuing requests: 4
- Number of servers processing them: 4
- Processing time for each request: 20 time units
- Client request generation period: 30, 20, 18 (i.e., workload: easy, tight and heavy; this last forcing some requests to be denied)

The first column displays the value of the processing time uncertainty, which ranges between 0 and 5% of the processing time of each request, i.e., between 0 and 1 time units. This is used by the system to assign to each request a deviation from its expected processing time, which simulates a more realistic situation where the actual processing times of requests are not perfect values but random variables. Of course, the heavier the workload the worst the results.

CRISP VERSION						
ProcTimeUnc	%Denied-30	%Overdue-30	%Denied-20	%Overdue-20	%Denied-18	%Overdue-18
0	0.00%	0.00%	0.00%	0.00%	7.61%	1.02%
0.1	0.00%	0.00%	0.00%	0.00%	7.61%	1.02%
0.2	0.00%	0.00%	0.00%	0.00%	7.61%	5.08%
0.3	0.00%	0.00%	0.00%	0.00%	6.53%	2.01%
0.4	0.00%	0.00%	0.00%	0.00%	7.07%	4.04%
0.5	0.00%	0.78%	0.00%	0.00%	6.53%	1.51%
0.6	0.00%	0.00%	0.00%	0.00%	7.07%	3.54%
0.7	0.00%	0.00%	0.00%	0.00%	6.53%	2.51%
0.8	0.00%	0.78%	0.00%	0.00%	7.61%	3.55%
0.9	0.00%	0.00%	0.00%	0.00%	8.00%	4.00%
1	0.00%	0.00%	0.00%	0.00%	7.07%	3.54%

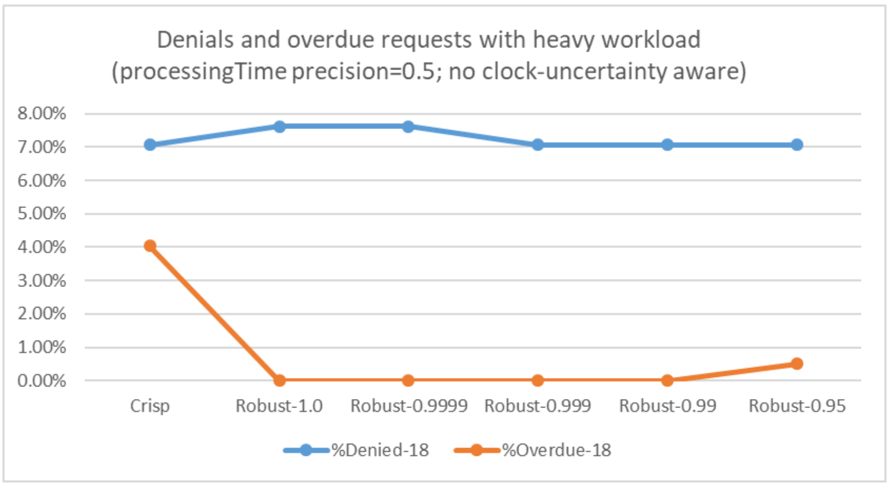


Then, we decided to manage the uncertainty due to the possible variations of the processing times of the requests using a robust control approach designed to function properly provided that values can vary. It takes a very conservative approach that uses intervals instead of values, and assumes a worst-case scenario. This is what we have called "Robust-1.0". To explore further scenarios, we also defined refinements of such a robust approach, where instead of using the worst-case interval, we just used the interval that ensures a certain precision in the comparison results. For this we assumed that the random variable of the processing time of requests followed a Normal distribution, and selected the number of standard deviations that ensured that the resulting interval contained such percentage of the values.

The results of these approaches are compared in the table below, in the case of heavy workloads. In the rest of the situations the system behaves well, modulo some spurious overdue values due to the imprecision of the clock.

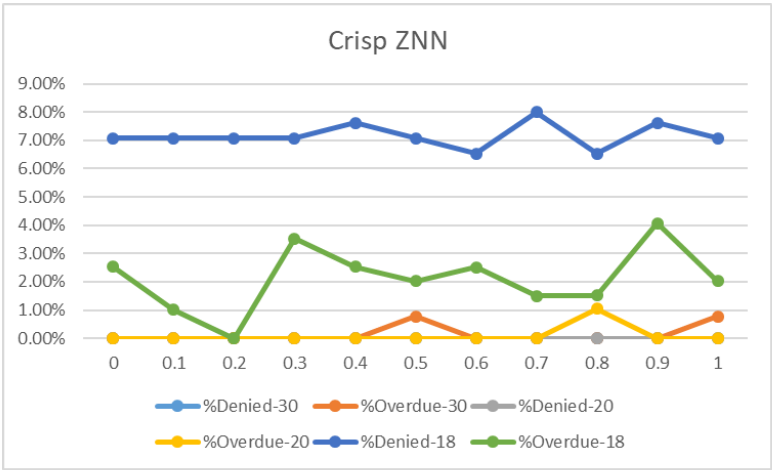
	%Denied-18	%Overdue-18
Crisp	7.07%	4.04%
Robust-1.0	7.61%	0.00%
Robust-0.9999	7.61%	0.00%
Robust-0.999	7.07%	0.00%
Robust-0.99	7.07%	0.00%
Robust-0.95	7.07%	0.51%

These results are shown in the chart below. We can see how robust solutions do not produce overdue responses, at the cost of more denials, as expected, because their margins are larger.



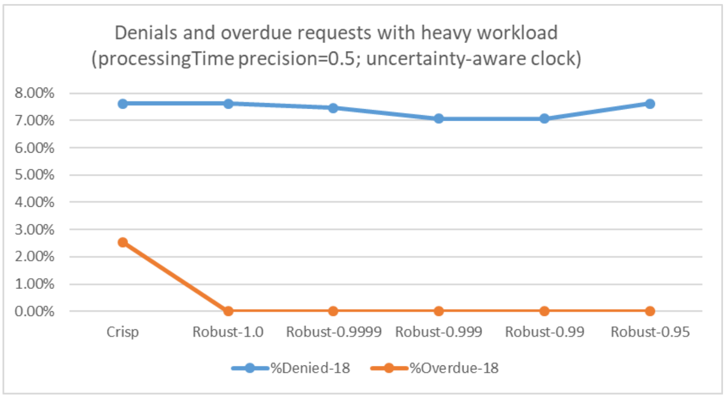
A second experiment aims at taming not only the uncertainty of the requests' processing times but also that of the clock. The results for the crisp solution are shown in the table and chart below. We can see how in this case the number of overdue requests is slightly less than in the previous case (the uncertainty of the clock was very small, though).

CRISP VERSION						
ProcTimeUnc	%Denied-30	%Overdue-30	%Denied-20	%Overdue-20	%Denied-18	%Overdue-18
0	0.00%	0.00%	0.00%	0.00%	7.07%	2.53%
0.1	0.00%	0.00%	0.00%	0.00%	7.07%	1.01%
0.2	0.00%	0.00%	0.00%	0.00%	7.07%	0.00%
0.3	0.00%	0.00%	0.00%	0.00%	7.07%	3.54%
0.4	0.00%	0.00%	0.00%	0.00%	7.61%	2.54%
0.5	0.00%	0.78%	0.00%	0.00%	7.07%	2.02%
0.6	0.00%	0.00%	0.00%	0.00%	6.53%	2.51%
0.7	0.00%	0.00%	0.00%	0.00%	8.00%	1.50%
0.8	0.00%	0.00%	0.00%	1.04%	6.53%	1.51%
0.9	0.00%	0.00%	0.00%	0.00%	7.61%	4.06%
1	0.00%	0.78%	0.00%	0.00%	7.07%	2.02%



What we can see is that in this case we completely avoid overdue requests in the robust versions. This is because we manage to successfully tame the two sources of uncertainty that were mentioned above: the imprecision of the processing times of requests, and the imprecision of the clocks.

	%Denied-18	%Overdue-18
Crisp	7.07%	2.02%
Robust-1.0	7.61%	0.00%
Robust-0.9999	7.61%	0.00%
Robust-0.999	7.07%	0.00%
Robust-0.99	7.07%	0.00%
Robust-0.95	7.07%	0.00%

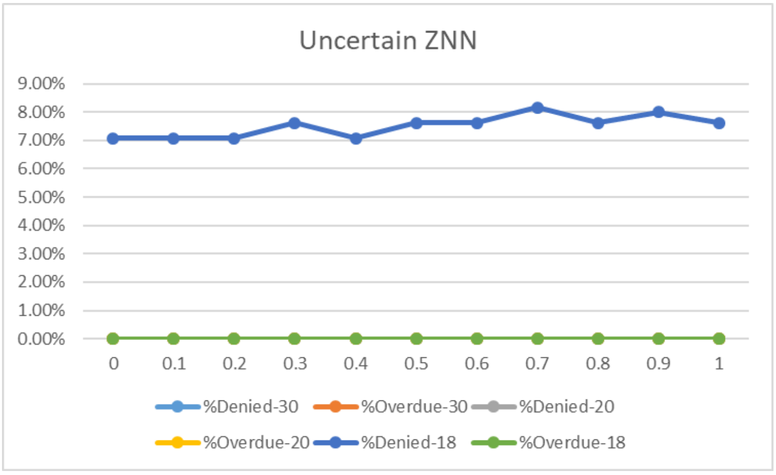


Uncertain values (file:RESULTS-ZNN.xlsx)

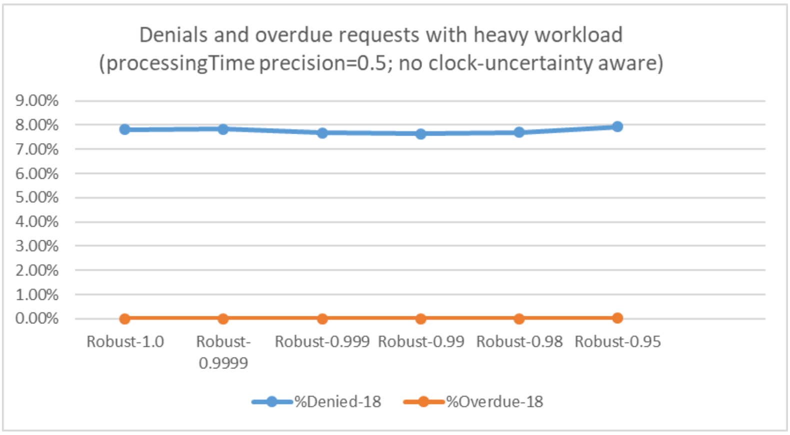
Here the results do not include crisp values, they are similar to the robust ones with intervals and degrees of confidence. The results are better now. No overdue requests in any case, and the denial rate is the same as in the previous cases (since it is mostly independent of the precision taken).

The table below shows the uncertain version with 1.0 robustness.

UNCERTAIN VERSION						
ProcTimeUnc	%Denied-30	%Overdue-30	%Denied-20	%Overdue-20	%Denied-18	%Overdue-18
0	0.00%	0.00%	0.00%	0.00%	7.07%	0.00%
0.1	0.00%	0.00%	0.00%	0.00%	7.07%	0.00%
0.2	0.00%	0.00%	0.00%	0.00%	7.07%	0.00%
0.3	0.00%	0.00%	0.00%	0.00%	7.61%	0.00%
0.4	0.00%	0.00%	0.00%	0.00%	7.07%	0.00%
0.5	0.00%	0.00%	0.00%	0.00%	7.61%	0.00%
0.6	0.00%	0.00%	0.00%	0.00%	7.61%	0.00%
0.7	0.00%	0.00%	0.00%	0.00%	8.16%	0.00%
0.8	0.00%	0.00%	0.00%	0.00%	7.61%	0.00%
0.9	0.00%	0.00%	0.00%	0.00%	8.00%	0.00%
1	0.00%	0.00%	0.00%	0.00%	7.61%	0.00%



	%Denied-18	%Overdue-18
Robust-1.0	7.80%	0.00%
Robust-0.9999	7.83%	0.00%
Robust-0.999	7.68%	0.00%
Robust-0.99	7.64%	0.00%
Robust-0.98	7.69%	0.00%
Robust-0.95	7.94%	0.02%


Denials and overdue requests with heavy workload
(processingTime precision=0.5; no clock-uncertainty aware)
Robust-1.0
Robust-0.9999
Robust-0.999
Robust-0.99
Robust-0.98
Robust-0.95
%Denied-18
%Overdue-18