Utrecht University

Master Thesis

Mathematical Sciences

Internet Service Time Distributions

Author:
Sjoerd Boersma

Supervisors:
Roberto Fernandez
Miroslav Zivkovic

July 31, 2015

Abstract

This thesis deals with the mathematics behind the speed of internet services and has both a theoretical and an experimental component. The goal is to give a better understanding of service time distributions and provide methods to predict its properties.

The theoretical part is written from an optimization perspective. It describes situations where a choice must be made from several service providers based on information about their service time distributions. This is done both for jobs that require a single service treatment and ones that require several services to work on them consecutively. It deals with problems such as incomplete information and service deterioration.

Subsequently we performed two series of experiments. The first consisted of measuring the response times for requests to actual internet services and analyzing the results. A comparison was made between several of the service time distributions and the effect of background traffic rate was shown by separating service times during the day from those at night.

The second series of experiments was done in a lab environment where facets like background traffic could be regulated. Distributions of response times were analyzed, and those with different background traffic rates compared. Extra attention was given to the tails of the distributions. An effort was made to create a model for service time distribution and their behaviour over time.

Acknowledgements

I would like to thank everyone who believed in me during my struggles with finishing my thesis. I have never stopped believing I would finish this thesis, even though it took me as long as it did. I particular my friends, parents and signficant others: thank for keeping up with me.

I would like to thank the people from TNO department 'Performance of Networks and Systems' during my internship. In particular Miroslav Zivkovic, who came up with the idea for this thesis and supervised me when I was there. Thanks for all your help. Also my second supervisor at TNO: Hans van den Berg and the department manager: Dick van Smirren were a big help and deserve credit.

I would like to thank Idilio Drago from the University of Twente, who set up the experiments and performed some of them, while also helping me doing the other ones myself. He went out of his way to support me without self-interest.

I would like to thank the people from the university who helped me in the making of this thesis. Cristian Spitoni, thanks for your remarks and examining as second reader. Karma Dajani, thanks for being my tutor and believing in me. Last but not least, thank you Roberto Fernandez for keeping up with me all those times I tried and even promised to finish my thesis but didn't. I owe you a lot for giving me the chance to make it all up and getting it done in the end.

Contents

Abstract **1**

Acknowledgements **2**

1 Introduction **5**

2 Single-service tasks **6**

2.1 Switching with 2 servers . . . 7

2.2 Switching with N servers . . . 10

2.3 Selection problems . . . 12

2.4 Incomplete info . . . 13

2.4.1 Two points . . . 13

2.4.2 More points . . . 16

2.4.3 Choosing a percentile . . . 17

2.5 Conclusions . . . 19

3 Multi-service networks **20**

3.1 Description . . . 20

3.2 Solution . . . 21

3.2.1 Look-up table . . . 22

3.3 Adding resources . . . 23

3.4 Example . . . 24

3.4.1 Last stage . . . 25

3.4.2 Middle stage . . . 26

3.4.3 First stage . . . 28

3.5 Conclusions . . . 29

4 Internet Experiments **31**

4.1 Pilot experiment . . . 31

4.2 Main experiment . . . 34

4.2.1 General distribution types . . . 36

4.2.2 Difference between day and night . . . 38

4.3 Conclusions . . . 40

5 Lab Experiments **43**

5.1 Motivation and Goals . . . 43

5.2 Setup . . . 43
5.3 Execution . . . 44
5.4 Analysis and Results . . . 45
5.4.1 Low background traffic . . . 45
5.4.2 Medium and high background traffic . . . 49
5.5 Notes on these results . . . 50
5.6 Conclusions . . . 52

6 Conclusions **54**
6.1 Further research . . . 54

References **55**

A Appendices **56**
A.1 Smoothing . . . 56
A.2 Rounding errors . . . 58

1 Introduction

In this thesis I deal with internet services and their distributions. I take a look at service selection for a single job when all or partial information is known in the section 'Single-service tasks'. I then generalize the problem to multiple stages/jobs in the section 'Multi-service networks'. The latter sections describe some experiments and results we did on the internet (in 'Internet Experiments') and in a controlled environment (in 'Lab Experiments'). These sections are preceded by this introduction and followed by conclusions and recommendations for further research. The sections all end with a conlusions part as well.

In the section 'Single-service tasks' I take a look at service tasks consisting of a single job. I first describe and solve a simple case where a job has to be finished before a deadline and we can choose from several service providers to perform it. I then look at some variations on this. The first variation is one where it is one allowed to switch to another service provider when performance of the first choice is dissapointing. Then I look at selection problems in general and applied to our case. Finally I make some remarks on how to deal with incomplete information, which is quite common in the world of internet services.

The next section, 'Multi-service networks', deals with a series of tasks that have to be performed in sequence. It describes a way to solve this problem using dynamic programming resulting in a look-up table. I then describe the factors that make the stages different from each other and which is most important to pay attention to. Also which one is most likely to benefit from adding resources. Finally I use an example with three stages and four service providers to get more insight in these matters and also make the case stronger.

The Internet Experiments described in the section by that name is meant to give us insights in the actual service and response time distributions we can encounter on the internet. A pilot experiment shows us the distributions usually have a general unimodal shape, which is partially confirmed by the main experiments, while also showing there are multimodal distributions possible. I investigate where differences may come from and how we can categorize the distributions. I take a look at differences between subdistributions taken by sorting the measurements on the time of the day they were taken, and conclude that busy services are probably slower than their counterparts that are used less often.

In the last section before the conclusions I describe a series of Lab Experiments we performed and the results I drew from analyzing them. I take an extensive look at the influence of background traffic on the service times and find such a link for both the mean service time and its variance. I fit the measured distributions to (double) normal ditributions an finally try to predict whether the results are applicable to the internet.

The various sections of this thesis build on results and remarks made in earlier sections, but it is quite possible to read the latter ones with limited knowledge of the earlier ones, the sections on the two series of experiments in particular

2 Single-service tasks

In this section I take a look at tasks that have to be executed by a single service provider ('server'). We try to finish the task before a given deadline. There may be more than one service provider at our disposal, but only one (at a time) may work on the job.

Consider the following model: a task is ought to be done within a given number of time units. If we manage to make this happen, we get a reward. If we fail, we have to pay a penalty. We have at our disposal a number of service providers numbered which have known service distributions. Each of them has to be paid for (if we use them). Only one service provider may work on the task at all time, but it may be allowed to switch to another service provider. However, if we choose to do so, all progress on the task will be lost and we also have to pay for the new server. Our goal is to maximize the expected revenue.

I will use the following notation throughout this section:

δ**:** deadline; number of time units available

r**:** reward in monetary units

v**:** penalty in monetary units

n**:** number of service providers (which are numbered $1 \dots n$)

X_i**:** service distribution of service provider i

f_i**:** probability density function of X_i

F_i**:** cumulative density function of X_i

$\bar{F}_i$**:** complementary cumulative distribution function of X_i $(= 1 - F_i)$

c_i**:** cost of service provider i in monetary units

$T_\cdot$**:** the total revenue distribution[1]

$\theta_\cdot$**:** switching time

$\theta_{\cdot *}$**:** optimal switching time

The deadline, reward, penalty and costs are all positive real numbers, the number of service providers is natural.

In the simplest case there is only one service provider at our disposal. In that case the following denotes our revenue distribution:

$$\mathbb{E}[T] = -c_1 + r \cdot \mathbb{P}[X_1 \leqslant \delta] - v \cdot \mathbb{P}[X_1 > \delta] = -c_1 + r \cdot F_1(\delta) - v \cdot \bar{F}_1(\delta) = r - c_1 - (r + v)\bar{F}_1(\delta).$$

Suppose we get the following choice: either use a given service provider or do nothing. In the latter case we automatically get the penalty v but avoid having to pay the cost of the service provider.

$$\begin{gathered}
\mathbb{E}[T] > -v \iff \\
r - c_1 - (r + v)\bar{F}_1(\delta) > -v \iff \\
(r + v) - c_1 - (r + v)\bar{F}_1(\delta) > 0 \iff \\
(r + v)F_1(\delta) > c_1.
\end{gathered}$$

[1] The total revenue as well as the switching time may depend on choices we make or conditions that hold. Those will be specified when they are introduced and indicated in the subscript.

We conclude that we should only use the service provider if its cost outweighs the expected revenue, which is given by the probability it is succesfull in doing the task before the deadline multiplied by the revenue we will get in that case plus the penalty we will miss. Also, we see that only the sum of the reward and penalty matters when we make our choice, not their respective values.

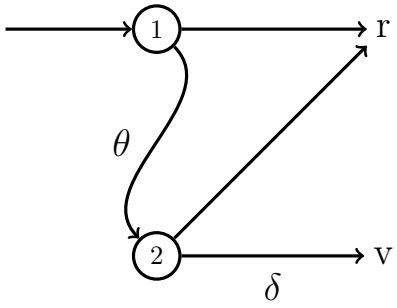
Now suppose we have to choose between two service providers and we have to stick with that one (we are not allowed to switch). On what basis should we make our choice? Of course we should choose the service provider that will give us the highest expected revenue. Let T_i denote the revenue distribution given that we choose server i. Then we get:

$$\begin{gathered}\mathbb{E}[T_1] > \mathbb{E}[T_2] \iff \\ r - c_1 - (r+v)\bar{F}_1(\delta) > r - c_2 - (r+v)\bar{F}_2(\delta) \iff \\ -c_1 - (r+v)\bar{F}_1(\delta) > -c_2 - (r+v)\bar{F}_2(\delta) \iff \\ c_1 + (r+v)\bar{F}_1(\delta) < c_2 + (r+v)\bar{F}_2(\delta).\end{gathered}$$

Again only the sum of the reward and the penalty matters, not their respective values. The preferable service provider is the one for which its cost, added to the probability it fails (to finish the task before the deadline) multiplied by the stake (the sum of reward and penalty) is the smallest.

2.1 Switching with 2 servers

Now consider the following configuration: there are two service providers at our disposal. The first one starts working on the task. Now it may happen that the deadline is near and the service is not yet finished. Then we now have the possibility to switch to the second server: we abort the service of the first server and request service from the second one, for which we have to pay as well. For certain service distributions this may result in a higher expected revenue. We assume it is not known how much advanced the service unless it is done, so the decision can only be based on the amount of time that has progressed and the amount of time until the deadline.



Why would we choose to do this ? It is in particular useful when we the first service provider is relatively slow and cheap, while the second one is faster but more expensive. We could have used the second service provider right away and be almost sure of meeting the deadline, but it would cost us the high costs of that service provider. By using the cheaper service provider first, we have a chance of finishing the task without paying the higher cost. It may also be profitable to switch if the service distributions are heavy-tailed[2].

Suppose we choose to switch whenever θ time units have passed. Let T_θ denote the revenue. If the service is finished by the first service provider before we switch, the total revenue is equal to: $T_\theta = r - c_1$. If we switch and the second server finishes the service before the deadline: $T_\theta = r - c_1 - c_2$. If we switch and the deadline is still not met: $T_\theta = -v - c_1 - c_2$. Then:

$$\begin{gathered}\mathbb{E}[T_\theta] = r - c_1 - c_2 \cdot \mathbb{P}[\text{server two is used}] - (r+v) \cdot \mathbb{P}[\text{deadline not met}] = \\ r - c_1 - c_2 \cdot \mathbb{P}[X_1 > \theta] - (r+v) \cdot \mathbb{P}[S_1 > \theta] \cdot \mathbb{P}[S_2 > \delta - \theta] = \\ r - c_1 - c_2 \cdot \bar{F}_1(\theta) - (r+v) \cdot \bar{F}_1(\theta) \cdot \bar{F}_2(\delta - \theta).\end{gathered}$$

[2]I will explain this term later, but in particular it results in a low probability of the service to be finished soon.

Let θ^* be the optimal switching time. Now $\mathbb{E}[T_\theta]$ is continuous in θ, so the optimal time to switch should be either 0, δ of a point in the open interval $(0, \delta)$. Since switching are time 0 and δ are nonsensical[3] I will not consider them. In that case the optimal switching time should be one which satisfies $\frac{d\mathbb{E}[T_\theta]}{d\theta}|_{\theta=\theta*} = 0$:

$$0 = \frac{d\mathbb{E}[T_\theta]}{d\theta}|_{\theta=\theta*} = c_2 \cdot f_1(\theta^*) + (r+v)(f_1(\theta^*) \cdot \bar{F}_2(\delta - \theta^*) - \bar{F}_1(\theta^*) \cdot f_2(\delta - \theta^*)).$$

This translates to:

$$h_1(\theta^*) \cdot \left[1 + \frac{c_2}{(R+V)(1 - F_2(\delta - \theta^*))}\right] = h_2(\delta - \theta^*),$$

where

$$h_i(t) = \frac{f_i(t)}{1 - F_i(t)}.$$

Given the previous system with two service providers, suppose they have an exponential service distribution. This case is in particular interesting, as it is memoryless. Let $X_i \sim \text{Exp}(\lambda_i)$. $\lambda_1 < \lambda_2$ and $c_1 < c_2$. As written before, this is an interesting case for switching: the first service provider is cheaper, the second one is faster. What is now the optimal switching time?

$$\begin{aligned}\mathbb{E}[T_\theta] &= r - c_1 - c_2 \cdot \bar{F}_1(\theta) - (r+v) \cdot \bar{F}_1(\theta) \cdot \bar{F}_2(\delta - \theta) \\ &= r - c_1 - c_2 \cdot e^{-\lambda_1 \theta} - (r+v) \cdot e^{-\lambda_1 \theta} \cdot e^{-\lambda_2(\delta - \theta)}.\end{aligned}$$

To maximize $\mathbb{E}[T_\theta]$, we differentiate to θ and set it equal to zero:

$$0 = \frac{d\mathbb{E}[T_\theta]}{d\theta}|_{\theta=\theta*} = c_2 \cdot \lambda_1 \cdot e^{-\lambda_1 \theta^*} + (r+v) \cdot e^{-\lambda_1 \theta^*} \cdot e^{-\lambda_2(\delta - \theta^*)} \cdot (\lambda_2 - \lambda_1).$$

Equivalently:

$$0 = c_2 \cdot \lambda_1 + (r+v) \cdot e^{-\lambda_2(\delta - \theta^*)} \cdot (\lambda_2 - \lambda_1).$$

This solves to

$$\theta^* = \delta - \frac{1}{\lambda_2} \cdot \log\left[\frac{(r+v)(\lambda_2 - \lambda_1)}{c_2 \lambda_1}\right]$$

In which cases will switching give us a higher profit than not switching at all? Let $\mathbb{E}[T_x]$ denote the expected profit in the non-switching case. Then:

$$\mathbb{E}[T_x] = r - c_1 - (r+v)\bar{F}_1(\delta).$$

Now switching at the best time is better than not switching whenever:

$$\mathbb{E}[T_x] < \max_\theta \mathbb{E}[T_\theta] = \mathbb{E}[T_{\theta*}].$$

This is equivalent to:

$$(r+v) \cdot \bar{F}_1(\delta) < c_2 \cdot \bar{F}_1(\theta^*) + (r+v)\bar{F}_1(\theta^*) \cdot \bar{F}_2(\delta - \theta^*)$$

If we revert to the exponential case again, we get:

$$(r+v) \cdot e^{-\lambda_1 \delta} < c_2 \cdot e^{-\lambda_1 \theta^*} + (r+v) \cdot e^{-\lambda_1 \theta^*} \cdot e^{-\lambda_2 \cdot (\delta - \theta^*)}$$

Or:

$$e^{-\lambda_1(\delta - \theta^*)} < \frac{c_2}{r+v} + e^{-\lambda_2(\delta - \theta^*)}.$$

[3] If you switch at time 0 not using the first service provider would be more profitable. Switching at time δ would only lead to extra costs and no additional chance of success

From last page is is known that:

$$\delta - \theta^* = \frac{1}{\lambda_2} \cdot \log\left[\frac{(r+v)(\lambda_2-\lambda_1)}{c_2\lambda_1}\right].$$

And our equation translates to:

$$\exp\left\{-\frac{\lambda_1}{\lambda_2} \cdot \log\left[\frac{(r+v)(\lambda_2-\lambda_1)}{c_2\lambda_1}\right]\right\} < \frac{c_2}{r+v} + \exp\left\{-\frac{\lambda_2}{\lambda_2} \cdot \log\left[\frac{(r+v)(\lambda_2-\lambda_1)}{c_2\lambda_1}\right]\right\}.$$

Or:

$$\left[\frac{c_2\lambda_1}{(r+v)(\lambda_2-\lambda_1)}\right]^{\lambda_1/\lambda_2} < \frac{c_2}{r+v} + \frac{c_2\lambda_1}{(r+v)(\lambda_2-\lambda_1)}.$$

Which can be rewritten as:

$$\left[\frac{c_2\lambda_1}{(r+v)(\lambda_2-\lambda_1)}\right]^{\lambda_1} < \left[\frac{c_2\lambda_2}{(r+v)(\lambda_2-\lambda_1)}\right]^{\lambda_2}.$$

Take note of the symmetry in the expression above. We can go further and rewrite:

$$\left[\frac{c_2\lambda_2}{(r+v)(\lambda_2-\lambda_1)}\right]^{\lambda_2} < \left[\frac{c_2\lambda_2}{(r+v)(\lambda_2-\lambda_1)}\right]^{\lambda_2-\lambda_1} \cdot \left[\frac{c_2\lambda_2}{(r+v)(\lambda_2-\lambda_1)}\right]^{\lambda_1}$$

and finally:

$$\left[\frac{\lambda_1}{\lambda_2}\right]^{\lambda_1} < \left[\frac{c_2\lambda_2}{(r+v)(\lambda_2-\lambda_1)}\right]^{\lambda_2-\lambda_1}.$$

What is the effect of having a second server to change to on our expected revenue? It is certainly not negative, as we can always choose not to use the second server. Under the condition that $\mathbb{E}[T_x] < \mathbb{E}[T_{\theta*}]$ adding a second server to the system is profitable. We'd like to know the extra profit we make, as a pecentage of $r+v$:

$$I := \frac{\mathbb{E}[T_{\theta*}] - \mathbb{E}[T_x]}{r+v}.$$

We've seen before that

$$\mathbb{E}[T_x] = -c_1 - V + F_1(\delta)(r+v)$$
$$\mathbb{E}[T_{\theta*}] = -c_1 - c_2 \cdot \bar{F_1}(\theta^*) - v + (F_1(\theta^*) + \bar{F_1}(\theta^*)F_2(\delta-\theta^*))(r+v)$$

It now follows that:

$$I = \frac{\mathbb{E}[T_{\theta*}] - \mathbb{E}[T_x]}{r+v} = \frac{-c_2 \cdot \bar{F_1}(\theta)}{r+v} + F_1(\theta) + \bar{F_1}(\theta)F_2(\delta-\theta) - F_1(\delta) =$$
$$(1-F_1(\theta)) \cdot \left[(F_2(\delta-\theta) - \mathbb{P}[\theta < X_1 \leqslant \delta]) - \frac{c_2}{r+v}\right].$$

The influence is thus equal to the rise in probability of succeeding because of the switching minus the extra costs for the switch (relative to the total revenue loss of failing), together discounted by the probability that the switching doesn't even occur.

I will consider the exponential case again. Let $X_i \sim \text{Exp}(\lambda_i)$. We have already seen that:

$$\theta^* = \delta - \frac{1}{\lambda_2} \cdot \log\left[\frac{(r+v)(\lambda_2-\lambda_1)}{c_2\lambda_1}\right]$$

is the optimal switching time. Now:

$$I = e^{-\lambda_1\theta^*} \cdot \left[e^{-\lambda_2(\delta-\theta^*)} - e^{-\lambda_1\delta} + e^{-\lambda_1\theta^*} - \frac{c_2}{r+v}\right].$$

2.2 Switching with N servers

We can expand the system of 2 servers with switching by adding more servers to the sequence. A sequence of N servers (denoted by $1, 2 \dots N$) is fixed and whenever server $i < N$ is working on the job, we can choose to abort its service and switch to server $i+1$. Let θ_i denote a switching time from server i to server $i+1$. Suppose they are fixed. Let $T|\theta_1 \dots \theta_{N-1}$ be the total revenue given these switching times. Then we get:

$$\mathbb{E}[T|\theta_1 \dots \theta_{N-1}] =$$

$$r \cdot \mathbb{P}[\text{deadline is met}] - v \cdot \mathbb{P}[\text{deadline is not met}] - \sum_{i=1}^{N} c_i \cdot \mathbb{P}[i^{th} \text{ server is used}]$$

$$= r - c_1 - (r+v) \cdot \mathbb{P}[\text{deadline is not met}] - \sum_{i=2}^{N} c_i \cdot \mathbb{P}[\text{not done by time } \theta_{i-1}]$$

Let $c_{N+1} := r + v$. Then the above equals:

$$r - c_1 - \sum_{i=2}^{N+1} c_i \cdot \mathbb{P}[\text{not done by time } \theta_{i-1}]$$

Now

$$\mathbb{P}[\text{not done by time } \theta_1] = \bar{F}_1(\theta_1).$$

Let $\theta_0 = 0$ and let α_i be the maximum time server i works on the job. Then $\alpha_i := \theta_i - \theta_{i-1}$ for $1 \leqslant i \leqslant N$.

$$\mathbb{P}[\text{not done by time } \theta_2] = \mathbb{P}[\text{not done by time } \theta_1] \cdot \mathbb{P}[\text{not done by time } \theta_2 | \text{not done by time } \theta_1] =$$

$$\bar{F}_1(\theta_1) \cdot \bar{F}_2(\theta_2 - \theta_1) = \bar{F}_1(\alpha_1) \cdot \bar{F}_2(\alpha_2)$$

Repeating this,

$$\mathbb{P}[\text{not done by time } \theta_i] = \prod_{k=1}^{i} \bar{F}_k(\alpha_k)$$

and thus:

$$\mathbb{E}[T|\theta_1 \dots \theta_{N-1}] = r - c_1 - \sum_{i=2}^{N+1} c_i \cdot \prod_{k=1}^{i-1} \bar{F}_k(\alpha_k) = r - c_1 - \sum_{i=1}^{N} c_{i+1} \cdot \prod_{k=1}^{i} \bar{F}_k(\alpha_k)$$

Calculating the optimal times for switching from one server to another is probably hard (or: time consuming) if we do it analytic with full information, and therefore we will look at a myopic formulation: the switching time θ_i from the i^{th} to the $(i+1)^{st}$ server is calculated under the assumption that the $(i+1)^{st}$ server is the last one. When the job has switched the horizon shifts to one extra server. As we have seen before, the optimal switching time θ_1 is the solution of the equation:

$$h_1(\theta_1) \left[1 + \frac{c_2}{(R+V)(1 - F_2(\delta - \theta_1))} \right] = h_2(\delta - \theta_1),$$

where

$$h_i(t) = \frac{f_i(t)}{1 - F_i(t)}.$$

For higher values of i, θ_i can be similarly obtained, since all past information is obsolete, only the current time matters, and it is forbidden to use information about possible future servers. Thus θ_i is the solution of the equation below, given that θ_{i-1} is already obtained. It follows from the fact that the situation is

equivalent to one where we start at time 0 and the deadline is at $\delta - \theta_{i-1}$. The value you'd find is now θ_{i-1} too high, so we must substract that. If we replace δ by $\delta - \theta_{i-1}$ and θ_i by $\theta_i - \theta_{i-1}$, we get:

$$h_i(\theta_i - \theta_{i-1})\left[1 + \frac{c_{i+1}}{(R+V)(1-F_{i+1}(\delta-\theta_i))}\right] = h_{i+1}(\delta - \theta_i),$$

given that the values for θ_i are increasing and in the interval $(0, \delta)$.

Now suppose we have exponential distributions again, $X_i \sim \text{Exp}(\lambda_i)$, while $\lambda_i < \lambda_j$ when $i < j$. Since we use the myopic strategy, we can use the following previous result:

$$\theta_1^* = \delta - \frac{1}{\lambda_2} \cdot \log\left[\frac{(r+v)(\lambda_2 - \lambda_1)}{c_2\lambda_1}\right].$$

Then:

$$(\theta_2^* - \theta_1^*) = (\delta - \theta_1^*) - \frac{1}{\lambda_3} \cdot \log\left[\frac{(r+v)(\lambda_3 - \lambda_2)}{c_3\lambda_2}\right]$$

or:

$$\theta_2^* = \delta - \frac{1}{\lambda_3} \cdot \log\left[\frac{(r+v)(\lambda_3 - \lambda_2)}{c_3\lambda_2}\right].$$

Similarly for all i

$$\theta_i^* = \delta - \frac{1}{\lambda_{i+1}} \cdot \log\left[\frac{(r+v)(\lambda_{i+1} - \lambda_i)}{c_{i+1}\lambda_i}\right].$$

If however for some i, $\theta_i^* > \theta_{i+1}^*$, we must find out whether it is better to switch anyway or not any more. After the optimal time switching decreases in power[4], so we either switch at time θ_i^* (so immediately, we don't use server $(i+1)$ even though we paid for it,) or we don't switch at all. As we have seen before switching is useful if and only if:

$$\mathbb{E}[T|\delta] < \max_\theta \mathbb{E}[T|\theta] = \mathbb{E}[T|\theta_i^*],$$

or:

$$e^{-\lambda_{i+1}(\delta-\theta_i)} < \frac{c_{i+2}}{r+v} + e^{-\lambda_{i+2}(\delta-\theta_i)}.$$

Suppose θ_1^* is a useful switching time ($0 < \theta_1^* < \delta$). Suppose $\theta_2^* < \theta_1^*$ as we calculated it. What is our best option, to take $\theta_2 = \theta_1^*$ or not switch at all? The former is the best option iff:

$$e^{-\lambda_2(\delta-\theta_1^*)} < \frac{c_3}{r+v} + e^{-\lambda_3(\delta-\theta_1^*)}.$$

Since θ_1^* is given above, we can put it in this formula and obtain:

$$\frac{c_2\lambda_1}{(r+v)(\lambda_2-\lambda_1)} < \frac{c_3}{r+v} + \left[\frac{c_2\lambda_1}{(r+v)(\lambda_2-\lambda_1)}\right]^{\lambda_3/\lambda_2}$$

or:

$$\left[\frac{c_2\lambda_1 - c_3\lambda_2 + c_3\lambda_1}{(r+v)(\lambda_2-\lambda_1)}\right]^{\lambda_2} < \left[\frac{c_2\lambda_1}{(r+v)(\lambda_2-\lambda_1)}\right]^{\lambda_3}$$

or:

$$\left[c_2\lambda_1 - c_3\lambda_2 + c_3\lambda_1\right]^{\lambda_2}\left[(r+v)(\lambda_2-\lambda_1)\right]^{\lambda_3-\lambda_2} < \left[c_2\lambda_1\right]^{\lambda_3}$$

[4]The optimal value of θ is the only value for which the derivative is zero. Since it is a maximum, $\mathbb{E}[T|\theta]$ must have a negative first derivative for θ larger than the optimum.

2.3 Selection problems

Consider the following problem:

A finite number N of concrete services is available for our job. These randomly put into a sequence. We are shown the details of the first server in the sequence, and must either choose or reject it. If we choose it, our job will be handled by this server and our revenue will be determined. If we reject the server we are shown the details for the second server are must choose whether to choose or reject it, and if we reject it we go on to the third server, etcetera. Once a server is rejected, we will thus never be able to reconsider it. Our goal is to maximize the expected revenue. The considerations are assumed to take no time.

If our goal was to maximize the probability of choosing the best one, while nothing was known of the distribution of possible expected revenues, this would be equivalent with the secretary or selection problem. However in this case a second best option is probably also a good outcome of the selection process. We assume it is also allowed to decide to select no service and just take a hit (pay the penalty).

It is important to understand that switching is not allowed. A deadline δ, revenue r and penalty v are known, while each concrete service a cost and a distribution are known. However as switching is not allowed, only the probability that the server will complete the job before the deadline is of interest. We can see:

$$\mathbb{E}[T_i] = (r+v)\mathbb{P}[\text{deadline is met}] - c_i - v = (r+v)F_i(\delta) - c_i - v.$$

The expected revenue of not choosing any concrete service is $-v$. If some server has an expected revenue smaller than $-v$, we will never choose it. In this model we will round all expected revenues smaller than $-v$ up to $-v$. The best possible concrete service would be one that has zero costs and a guarantee that the deadline is met. For such a service the expected revenue is r.

We will assume the services are sequences at random, but how are their details determined? We will assume these details are i.i.d. according to some distribution functions. Define $G(x) = \mathbb{P}[\mathbb{E}[T_1] \leqslant x]$ for $-v \leqslant x \leqslant r$ and let $g(x)$ be the associated PDF. Depending on this distribution function, we may define a strategy that maximizes our expected reward. If the distribution is uniform, by scaling the problem we get a uniform distribution on $(0,1)$, a case that has been solved by John Gilbert and Frederick Mosteller in 1966 ("Recognizing the maximum of a sequence"). If nothing is known about the distribution we can still choose the best server with probability $\frac{1}{e}$.

Suppose X_i $(1 \leqslant i \leqslant N)$ are i.i.d. with law X. Let $q_X(t)$ be the expected revenue if we are allowed to choose from t servers which are ditributed as X. If we consider server X_{N-t}, we should choose it if it's value is larger than $q_X(t)$, and not if it's smaller. Thus we can determine $q_X(t)$ iteratively:

$$q_X(t+1) = \mathbb{P}[X > q_X(t)]\mathbb{E}[X|X > q_X(t)] + \mathbb{P}[X < q_X(t)]q_X(t) =$$
$$\bar{F}_X(q_X(t))\frac{\int_{q_X(t)}^{\infty} x f_X(x)dx}{\int_{q_X(t)}^{\infty} f_X(x)dx} + F_X(q_X(t))q_X(t) =$$
$$F_X(q_X(t))q_X(t) + \int_{q_X(t)}^{\infty} x f_X(x)dx = \int_0^{\infty} \max[q_X(t), x] \cdot f_X(x)dx.$$

Since the last server has to be chosen if no other choice has been made yet, $q_X(1) = \mathbb{E}[X]$ (or similarly $q_X(0) = 0$).

For instance, if $X \sim U(0,1)$,

$$q_X(t+1) = q_X(t)^2 + \frac{(1+q_X(t))(1-q_X(t))}{2} = \frac{1}{2}(1+q_X(t)^2).$$

Now $q_X(1) = \frac{1}{2}$, $q_X(2) = \frac{5}{8}$, $q_X(3) = \frac{89}{128}$, $q_X(4) = \frac{24.305}{32.768}$. Now from the non-negativity of squares it follows that

$$(q_X(t) - 1)^2 \geqslant 0 \Rightarrow q_X(t) \leqslant \frac{1}{2}(1 + q_X(t)^2).$$

Where equality only holds when $q_X(t) = 0$, and since also $\frac{1}{2}(1+q_X(t)^2) < 1$ if $q_X(t) < 1$, the series $(q_X(t))_{t\in\mathbb{N}}$ grows with one as its upper bound, with

$$\lim_{n\to\infty} q_X(n) = 1.$$

If $X \sim U(0, a)$, with some scaling we get that $q_X(t)$ is just a times larger than in the case above, so

$$\frac{q_X(t)}{a} = \frac{1}{2}(1 + \left(\frac{q_X(t)}{a}\right)^2)$$

Or:

$$q_X(t) = \frac{1}{2}\left(a + \frac{q_X(t)^2}{a}\right).$$

If $X \sim U(-v, r)$, we have to add v to the solution for $U(0, r + v)$, so:

$$q_X(t) + v = \frac{1}{2}\left((r + v) + \frac{(q_X(t) + v)^2}{r + v}\right).$$

Or:

$$q_X(t) = \frac{1}{2}\left(r - v + \frac{q_X(t)^2 + 2V \cdot q_X(t) + V^2}{r + v}\right).$$

The natural outcome of this project is however not an expected revenue (although you will need it for the calculation of the thing that follows) but mainly a prescription as to whether we should choose a server or not. This however simply follows from the work above:

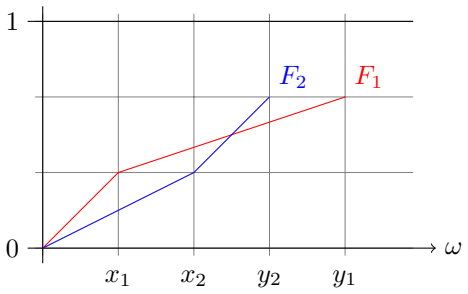
If there are N servers, and the i^{th} is being considered. Select it if $X_i \geqslant q_X(N - i)$, and otherwise reject it.

2.4 Incomplete info

Unless we actually measure the distribution of a service provider, we are unlikely to know it. If we make an agreement to use a service with a service provider, it will most likely include a Service Level Agreement, including some percentiles of the service. Suppose we have a few of those and need to decide which service we prefer based on only that, how can we make a good decission?

2.4.1 Two points

Consider two concrete services S_1 and S_2, with corresponding costs c_i, CDF's F_i and PDF's f_i. However the CDF and PDF are mostly unknown. There are two levels H (high) and L (low) between zero and one, $L < H$. Now x_1, x_2, y_1, y_2 are known for which: $F_i(x_i) = L$ and $F_i(y_i) = H$. Of course $x_i < y_i$, but to make things interesting, $x_1 < x_2$ while $y_1 > y_2$.



Suppose there is a deadline δ and you have the choice to let either of the two servers work on the job until it finishes or the deadline expires. No switching is allowed and we want to maximize our expected revenue. First consider the case where $c_1 = c_2$, and compare the two servers solely on their serving performance. Given δ, we will choose the server that has the largest probability to finish the service before that deadline.

If $x_1 \leqslant \delta \leqslant x_2$, $F_2(\delta) \leqslant L \leqslant F_1(\delta)$, so we should choose to use server one. Similarly if $y_2 \leqslant \delta \leqslant y_1$, $F_1(\delta) \leqslant H \leqslant F_2(\delta)$ and we should choose server 2. For all other values of δ, both $F_1(\delta) < F_2(\delta)$ and $F_2(\delta) < F_1(\delta)$ are possible. How to deal with this?

It is possible to create a set of random variables $X(\omega)$ that denotes the possible values of $F(\omega)$, but how can to decide on a distribution? Since no further information is available, we could decide on a deterministic function for which the PDF of an interval between two known point is constant. The PDF we thus construct is denoted by f_i'. Then:

$$f_i'(\omega) = \begin{cases} \frac{L}{x_i} & \text{if } \omega < x_i \\ \frac{(H-L)}{y_i - x_i} & \text{if } x_i < \omega < y_i \\ \text{unknown} & \text{if } y_i < \omega \end{cases} \tag{2.1}$$

The corresponding CDF, F_i', is given by:

$$F_i'(\omega) = \begin{cases} \frac{\omega \cdot L}{x_i} & \text{if } \omega < x_i \\ L + \frac{(\omega - x_i)\cdot(H-L)}{y_i - x_i} & \text{if } x_i < \omega < y_i \\ \text{unknown} & \text{if } y_i < \omega \end{cases} \tag{2.2}$$

If we now define $z_i = y_i - x_i$ and $J = H - L$, for $x_i < \omega < y_i$: $f_i'(\omega) = \frac{J}{z_i}$, $F_i'(\omega) = L + \frac{J \cdot (\omega - x_i)}{z_i}$.

Since $x_1 < x_2$, for $\omega < x_1$: $F_1'(\omega) = \frac{\omega \cdot L}{x_1} > \frac{\omega \cdot L}{x_2} = F_2'(\omega)$, and thus we should choose server one for these smaller values of the deadline.

For values of ω larger than y_1 it is hard to say something, since these points lie further than our largest data point. However if we must choose, since we know that $F_2(y_1) \geqslant F_2(y_2) = F_1(y_1)$ and thus the second server is most likely to be the best choice.

The last and most interesting area is $x_2 < \omega < y_2$. If ω is one of the smaller points in this interval it is most likely that server 1 is the best choice, but closer to y_2 server 2 is better. The tipping point is ω for which $F_1(\omega) = F_2(\omega)$, or:

$$L + \frac{(\omega - x_1)\cdot(H-L)}{y_1 - x_1} = L + \frac{(\omega - x_2)\cdot(H-L)}{y_2 - x_2}.$$

This is equivalent to:

$$z_2(\omega - x_1) = z_1(\omega - x_2),$$

and solves to:

$$\omega = \frac{z_1 x_2 - z_2 x_1}{z_1 - z_2} \ (=: \delta^*)$$

For the given situation, and assuming the methods above wordly, we conclude that server 1 should be chosen whenever $\delta \leqslant \delta^* = \frac{z_1 x_2 - z_2 x_1}{z_1 - z_2}$, and otherwise server 2 is the best choice.

What if $c_1 \neq c_2$? If server 1 is cheaper than server 2, the tipping point will be larger than in the section above, or even be infinite (so we always choose server 1). If server 2 is the cheapest, the tipping point will be smaller or even equal to zero (so we always choose server 2). I turns out there may be more tipping points (for instance, if the second server is slightly cheaper, it is the best option for large deadlines due to better performance there, but also for really small deadlines, because it is cheaper and probably neither of the two servers will finish the job by then).

Let $T_i(\omega)$ denote the revenue when the deadline is given to be ω. Then for $\delta < x_i$:

$$\mathbb{E}[T_i(\delta)] = -c_i - v + (r+v) \cdot F_i'(\delta) = -c_i - v + \frac{(r+v) \cdot \delta \cdot L}{x_i}$$

For $x_i < \delta < y_i$:

$$\mathbb{E}[T_i(\delta)] = -c_i - v + (r+v) \cdot F_i'(\delta) = -c_i - v + (r+v) \cdot \left(L + \frac{(\delta - x_i) \cdot (H-L)}{y_i - x_i} \right)$$

First consider deadlines before x_1. If $c_1 < c_2$, we should choose server 1 for sure, but if $c_2 < c_1$, it depends on the difference. We should choose server 2 iff:

$$\mathbb{E}[T_1(\delta)] \leqslant \mathbb{E}[T_2(\delta)] \iff -c_1 + \frac{(r+v) \cdot \delta \cdot L}{x_1} \leqslant -c_2 + \frac{(r+v) \cdot \delta \cdot L}{x_2} \iff$$

$$\delta \leqslant \frac{c_1 - c_2}{L(r+v)(\frac{1}{x_1} - \frac{1}{x_2})} = \frac{x_1 x_2 (c_1 - c_2)}{L(r+v)(x_2 - x_1)}$$

So for small deadlines we should choose server 2 as it is cheaper and neither of the servers has a good probability to complete the service in time anyway (in reality we would probably choose neither for the smallest values, but that is not an option that is considered now). When the deadline increases, the better performance of server 1 will become more important. Server 1 will never be chosen for these small values if the expression above (equals or) exceeds x_1. This happens when:

$$x_1 \leqslant \frac{x_1 x_2 (c_1 - c_2)}{L(r+v)(x_2 - x_1)},$$

or equivalently:

$$L \cdot (1 - \frac{x_1}{x_2}) \leqslant \frac{c_1 - c_2}{r+v}.$$

We see here that server 2 is mostly interesting for early deadlines if the relative difference between x_1 and x_2 is small and/or the difference in costs is large, when compared to the penalty and the reward. These agree with our common sense.

Now let us consider a deadline between x_1 and x_2. Again if server 1 is cheaper we should choose it, as it performs better. Suppose $c_2 < c_1$. In that case we should choose server 2 iff:

$$\mathbb{E}[T_1(\delta)] \leqslant \mathbb{E}[T_2(\delta)] \iff$$

$$-c_1 + (r+v) \cdot \left(L + \frac{(\delta - x_1) \cdot (H-L)}{y_1 - x_1} \right) \leqslant -c_2 + \frac{(r+v) \cdot \delta \cdot L}{x_2}$$

This expression is harder to solve. In general we can just write that we must choose server 2 iff:

$$\mathbb{E}[T_1(\delta)] \leqslant \mathbb{E}[T_2(\delta)] \iff F_1'(\delta) - F_2'(\delta) \leqslant \frac{c_1 - c_2}{r + v},$$

which could be read as: the difference in probability of finishing the job must outweigh the difference in costs, weighed to the sum of penalty and reward, where the last one is of course the difference between revenue when failing and when succeeding.

Let us look at deadlines between x_2 and y_2. Then we should choose server 2 iff:

$$\left(L + \frac{(\delta - x_1) \cdot (H - L)}{y_1 - x_1}\right) - \left(L + \frac{(\delta - x_2) \cdot (H - L)}{y_2 - x_2}\right) \leqslant \frac{c_1 - c_2}{R + V},$$

2.4.2 More points

What if the distribution of the r.v. X of service time of server is not known, but all information we get is a few data point of the form (a_i, b_i) for $1 \leqslant I \leqslant n$ which inform us that $\mathbb{P}[X \leqslant b_i] = a_i$, or otherwise (if they take the form of a guarantee, a minimal promised performance), $\mathbb{P}[X \leqslant b_i] \geqslant a_i$. We will consider the first of these two options. Suppose $a_j < a_k$ and $b_j < b_k$ whenever $j < k$. If these data point are given, we have a few options:

- Consider the worst case scenario, for which the CDF is the infimum of all possible CDF's: all data point are as prescribed and for all other point $F(x) = \max_{i:b_i \leqslant x} a_i$, so that $\lim_{x \uparrow b_i} = a_{i-1}$. Thus $\mathbb{P}[X = b_i] = a_i - a_{i-1}$ while $\mathbb{P}[X \neq b_i \forall i] = 0$. In that case the only reasonable moments to switch are the b_i (given that you only switch if the service is not done right in that moment), since inbetween two b_i's there is no possibility that the service finished, and neither will any new information become available. In particular cases it is possible that switching a small amount of time later than some b_i is equally good as switching precisely at the b_i, but it will never be better.
- In the best case scenario $F(x) = \min_{i:b_i \geqslant x} a_i$. This is similar to the case above, but makes little sense in the conventional way since it is not right-continuous. If we define G as the infimum of all CDF's that are bigger than or equal to F on all points, we will get $G(x) = \min_{i:b_i > x} a_i$. This is almost equal to F, only at the b_i's it is different, so it does not satisfy the only constraint we gave the CDF...
- The linear case: suppose we have sufficiently enough data points and assume that some continuity applies to our CDF (this is in particular true if it's PDF is continuous or even just a conventional PDF where $\mathbb{P}[X = x] = 0$ for all particular values of x). Then we may get a good approximation to the actual CDF if we use linear interpolation. So for $x \in (b_i, b_{i+1})$:

$$F(x) = a_i + \frac{x - b_i}{b_{i+1} - b_i}(a_{i+1} - a_i) = \frac{x - b_i}{b_{i+1} - b_i} a_{i+1} + \frac{b_{i+1} - x}{b_{i+1} - b_i} a_i$$

In this case $f(x) = \frac{a_{i+1} - a_i}{b_{i+1} - b_i}$ for all these x, so

$$f = \sum_{i=1}^{n} 1_{[b_{i-1}, b_i)} \frac{a_{i+1} - a_i}{b_{i+1} - b_i},$$

Where $b_0 = 0$. If $a_n = 1$ this function is a probability density function, otherwise it misses a bit at the end.

We can only get partial knowledge of the distribution, but can choose the percentiles. How should we choose?

If a sequence of point on the graph of the CDF is given, how could we measure the uncertainty in the CDF? And how many points do we need to get this uncertainty below a level that is acceptable? The main

reason that we want to avoid uncertainty is that with higher uncertainty the probability of us making wrong decisions (as of whether to switch) is also higher, and bad decisions cost us revenue. But how to measure it. Let us define the distance between two CDF's as the integral over the absolute value of their values:

$$d(F,G) = \int |F-G| d\lambda = \int_{F<G} (G-F) d\lambda + \int_{G<F} (F-G) d\lambda \tag{2.3}$$

That way we measure the amount of 'displaced' probability. Now let us take the uncertainty to be the maximum of $d(F,G)$ where F and G are CDF's which comply to the data points:

$$u((a_i,b_i)_i) = \sup_{F,G \in \Omega((a_i,b_i)_i)} d(F,G),$$

where $\Omega((a_i,b_i)_i)$ is the set of CDF's that satisfy the data point $((a_i,b_i)_i)$. The supremum is obtained by taking the difference between the infimum and the supremum of all possible CDF's, which have been described before. It follows that:

$$u((a_i,b_i)_i) = \sum_{i=0}^{n} (a_{i+1}-a_i)(b_{i+1}-b_i)$$

Where we define $a_0 = b_0 = 0$, $a_{n+1} = 1$, $b_{n+1} = D$, where D could be chosen to be the overal deadline δ or have some arbitrary smaller value.

Suppose we have two consecutive data point (a,b) and (c,d). The uncertainty generated by these two data point is equal to $(c-a)(d-b)$. What happens if we add an extra data point (e,f) in between? Of course that depends on the choice of (e,f). If we choose it to be really close to one of the other data points, the uncertainty will decrease only slightly. If we however choose (e,f) to be far away from the other data point, more could be gained. How much? At least 50%. If we choose $e = \frac{a+c}{2}$, we get the uncertainty $(c-e)(d-f) + (e-a)(f-b)$ is exactly equal to half the earlier uncertainty. The same is true if we choose (e,f) such that $f = \frac{b+d}{2}$. If the actual CDF turns out to be linear between (a,b) and (c,d), this is the best we can do. In other cases where the CDF goes through the point $(\frac{a+c}{2}, \frac{b+d}{2})$, this is also true. However if the CDF crosses the line segment from (a,d) to (b,c) in some other point (and it must cross that segment somewhere) we can do better. In particular that is the case for this intersection point. The point on the segment have the property that they are of the form $(\lambda a + (1-\lambda)c, \lambda b + (1-\lambda)d)$ for some λ. If we take $\lambda \in [0,1]$ we get exáctly the line segment. So for given λ the uncertainty generated by the three point is:

$$(\lambda a + (1-\lambda)c - a)(\lambda b + (1-\lambda)d - b) + (c - \lambda a - (1-\lambda)c)(d - \lambda b - (1-\lambda)d)$$

$$= (1 - 2\lambda - 2\lambda^2)(c-a)(d-b).$$

To evaluate the performance we must look at the values $k(\lambda) := (1 - 2\lambda - 2\lambda^2)$ will take. For $\lambda = \frac{1}{2}$ k equals $\frac{1}{2}$. If we differentiate k to λ and set it to be zero $\lambda = \frac{1}{2}$ is again the (only) answer. As some other values of λ give lower values of k, $\lambda = \frac{1}{2}$ is a maximum, and the performance in all other point on the anti-diagonal is strictly better.

Although we have seen that choosing the crossing of the anti-diagonal and the CDF is a good one (with a performance guarantee), it need not be the best one.

2.4.3 Choosing a percentile

If it was allowed to choose which percentiles were agreed upon in the SLA, what should we choose? Let us look at the case where only one percentile is allowed.

Suppose a CDF is given to you, and not to some other person. You are allowed to give this other person one data point besides two he already knows: $(0,0)$ and (a,δ), where your data point of the form (b,c) should satisfy $0 < c < \delta$ and $F(c) = b$. How can you do this and minimize the uncertainty function for the other person?

For given $(F(c),c)$, $u(\ldots) = cF(c) + (\delta - c)(a - F(c))$. If F is continuous, the maximum is attained at a point where $\frac{du}{dc}$ is zero. If it is not continuous, which implies there are point with a nonzero probability, it could also be a point on which the CDF is not continuous. We will consider the first:

$$0 = \frac{du(c)}{dc} = F(c) + cf(c) - (a - F(c)) - (\delta - c)f(c) = 2F(c) - a + (2c - \delta)f(c)$$

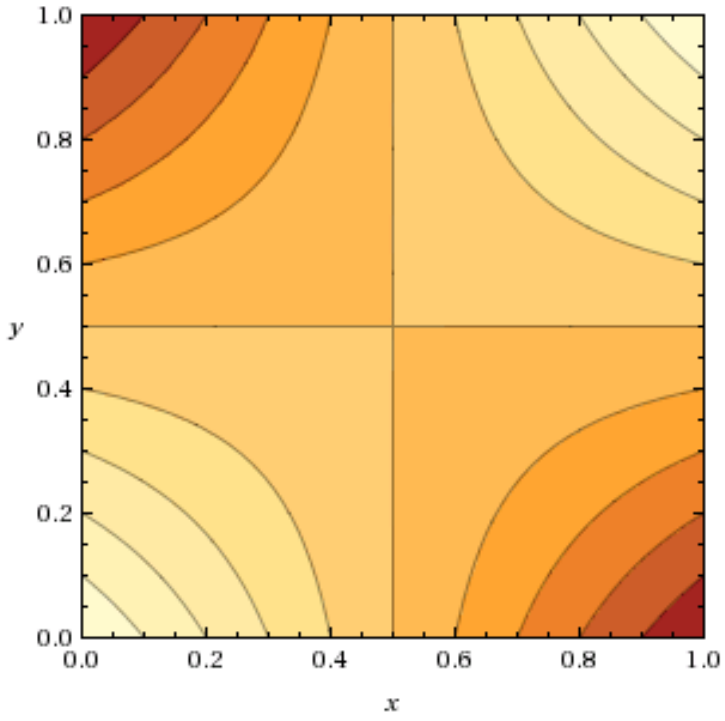
If we take a uniform distribution ($f(x) = \frac{a}{\delta}$, $F(x) = \frac{ax}{\delta}$), we get $c = \frac{\delta}{2}$.

For other distributions the calculations are much harder to do. If we consider a exponential distribution with rate λ, $f(x) = \lambda e^{-\lambda x}$, $F(x) = 1 - e^{-\lambda x}$, $a = F(\delta) = 1 - e^{-\lambda\delta}$. Then we get:

$$0 = 2 * (1 - e^{-\lambda c}) - (1 - e^{-\lambda\delta}) + (2c - \delta)\lambda e^{-\lambda c} = 1 + e^{-\lambda\delta} + (2\lambda c - \lambda\delta - 2)e^{-\lambda c}.$$

Using Wolfram|Alpha on the internet[5] I generated the following contourplot of the function

$$u((0,0),(x,y),(1,1)) = xy + (1 - x)(1 - y) = 1 + 2xy - x - y.$$



In this picture darker shades are better positions for our extra data point. If we take other values for the lower en upper bounds the picture will be the same, but stretched out. As we've seen before the horizontal line and the vertical line halfway have the property that here the gain is 50%. Again this is only certainly our best possibility if the CDF goes through the central point.

[5] contourplot $1 + 2xy - x - y$ for $\{x,0,1\}, \{y,0,1\}$

2.5 Conclusions

Given a job that has deadline δ and we have to choose one service provider which will perform it, earning r if we succeed and losing v if we don't. If the service providers are indexed by i and have CDF F_i and cost c_i, we should minimize $c_i + (r+v)\bar{F}_i(\delta)$ in order to maximize our revenue. Here $r+v$ is the value we lose if we don't succeed compared to meeting the deadline, while $\bar{F}_i(\delta)$ is the possibility of that happening.

If we have the possibility to switch to another service provider if the first one disappoints, we can calculate the optimal switching time (if it exists), but we did not find a (simple) analytical solution in general. For more than 2 service providers calculations can be made easier with potentially godd results by only considering two service providers at a time.

When we have to select a service out of N without going back, we should choose the i^{th} when it is presented to us if: $X_i \geqslant q_X(N-i)$, and otherwise reject it. These values can be calculated when it is known what the distribution of expected revenues for the service providers is.

We have to be able to deal with incomplete info, as those are in practice caused by the Service Level Agreements (SLAs) consisting of percentiles which are customary in the industry of internet services. To gain the maximal amount of information, one should choose to agree upon percentiles that are as far apart as possible, even though it is customary to get only high percentiles.

3 Multi-service networks

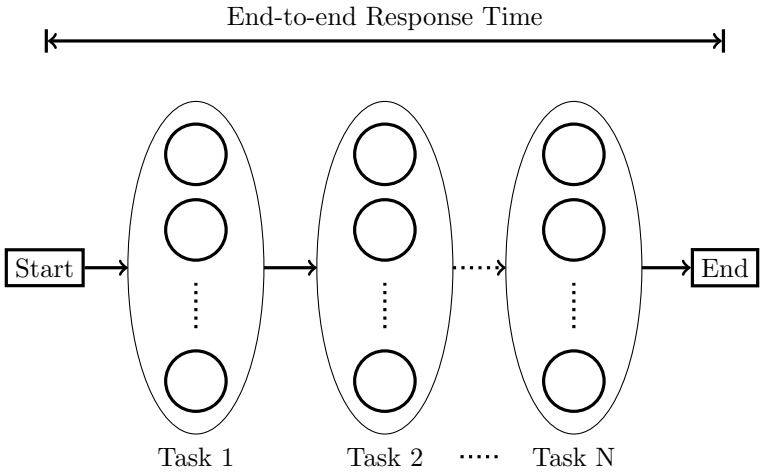
In this section I will take a look at multi-service networks, in which several tasks have to be performed sequentially before a deadline. The first task has to be executed first, then the second one etcetera.

3.1 Description

A customer has a project that consists of several tasks. We offer to carry out the project by executing the tasks sequentially, one at a time. If we manage to finish all tasks before a specified deadline, we receive a reward from the customer. If we fail, we have to pay a penalty to the customer.

Now for each task, instead of executing it ourself, we try to find a third party to perform it for us, in exchange for some money. In each stage may have a variety of choices for this contractor party, and choose between them based on their price and performance. In contrast to before, switching is not allowed.

Our overal goal is to maximize our expected revenue, which consists of the expected reward minus the expected penalty and costs of the contractor parties.



In this section I will use the following notation:

δ**:** The *end-to-end deadline* is the amount of time units in which all tasks should be completed: $\delta \in \mathbb{R}^+$.

r**:** The *reward* is a monetary value that is awarded to us if we manage to meet the end-to-end deadline: $r \in \mathbb{R}^+$

v**:** The *penalty* is a monetary value that we have to pay if we fail to meet the end-to-end deadline: $v \in \mathbb{R}^+$.

n**:** The *number of tasks*, also called *stages*: $n \in \mathbb{N}$.

k_i**:** The *number of service providers* in stage i: $k_i \in \mathbb{N}$.

$s_{i,j}$**:** The j^{th} *service provider* in stage i.

$X_{i,j}$**:** The *service time distribution* of $s_{i,j}$.

$F_{i,j}$**:** The *cumulative distribution function* (CDF) of $X_{i,j}$: $F_{i,j}(t) = \mathbb{P}[X_{i,j} \leqslant t]$.

$f_{i,j}$**:** The *probability distribution function* (PDF) of $X_{i,j}$: $f_{i,j}(t) = \frac{d}{dx} F_{i,j}(x)|_{x=t}$.

$c_{i,j}$**:** The *cost* of invocation of $s_{i,j}$, the price we have to pay if we use it.

For all of the above: $i \in \{1, 2, \ldots, n\}$ and $j \in \{1, 2, \ldots, k_i\}$. Let the following be defined:

$\mathcal{L}$**:** A *policy* which assigns to each stage i and time $t \in [0, \delta]$ a server $\mathcal{L}(i, t)$. It is thus a random assignment, where the assignment in a stage depends on the service times in earlier stages.

T**:** The *total revenue* is the reward (if it is awarded to us) minus the penalty (if we have to pay it) minus the costs of the invoked services, given a policy.

θ_i**:** The time left after stage i, in which the stages $(i+1) \ldots n$ must be completed in order to meet the deadline, $\forall\ i \in \{1, \ldots, n\}$.

Now

$$\theta_0(\mathcal{L}) = \delta \text{ and } \theta_i(\mathcal{L}) = \theta_{i-1}(\mathcal{L}) - X_{i,\mathcal{L}(i,\theta_{i-1})}$$

$$\text{Equivalently } \theta_i(\mathcal{L}) = \delta - \sum_{k=1}^{i} X_{k,\mathcal{L}(k,\theta_{k-1})}$$

$$\{\sum_{k=1}^{n} X_{k,\mathcal{L}(i,\theta_{i-1})} \leqslant \delta\} = \{\theta_n(\mathcal{L}) \geqslant 0\}.$$

$$T(\mathcal{L}) = r \cdot \mathbb{1}_{\{\theta_n(\mathcal{L}) \geqslant 0\}} - v \cdot \mathbb{1}_{\{\theta_n(\mathcal{L}) < 0\}} - \sum_{k=1}^{n} c_{k,\mathcal{L}(i,\theta_{i-1})}.$$

Now the objective is to find a policy $\mathcal{L}^*$ such that $\mathbb{E}[T(\mathcal{L}^*)]$ is maximized:

$$\mathcal{L}^* = \arg\max_{\mathcal{L}} \mathbb{E}[T(\mathcal{L})],$$

where

$$\mathbb{E}[T(\mathcal{L})] = (r + v) \cdot \mathbb{P}[\theta_n(\mathcal{L}) \geqslant 0] - v - \mathbb{E}[\sum_{i=1}^{n} c_{i,\mathcal{L}(i,\theta_{i-1})}].$$

3.2 Solution

If there is full information on the performance of the contract parties, in the sense that the probability distributions of their execution times are known, there is an optimal solution, which could be calculated. This can be done using dynamic programming, although it takes a lot of time and only approximates the optimal solution. This will be sufficient if the distributions are time-invariant (do not change) and known in advance.

The dynamic programming solution works as follows: for each contractor we consider the probability density function of their service distribution, or if it is not given our best guess of the probability density function (which may be based on observations from the past or promises by the contractor). To make calculations possible we transform it to a discrete function, which is similar to the continuous function we considered. Thus if f is the probability density function, and let $(c_i)_{0 \leqslant i \leqslant M}$ be an increasing series with $c_0 = 0$ and $c_M = D$ (the deadline), let g be the discreticized version of f:

$$g = \sum_{i=1}^{M} a_i \delta_{b_i},$$

where δ_x denotes the Dirac measure in point x, $b_i = \frac{1}{2}(c_{i-1} + c_i)$ and $a_i = \int_{c_{i-1}}^{c_i} f(x)dx$. So:

$$g = \sum_{i=1}^{M} \int_{c_{i-1}}^{c_i} f(x)dx \cdot \delta_{\frac{1}{2}(c_{i-1}+c_i)}.$$

Now we first calculate for the last stage then the one before that etcetera and make a look-up table.

3.2.1 Look-up table

Let there be N stages and let the i^{th} stage have K_i concrete services: $C_{i,1} \dots C_{i,K_i}$. Let $f_{i,j}$ and $F_{i,j}$ be the initial PDF and CDF of $C_{i,j}$, respectively, and $c_{i,j}$ the costs to invoke it. Let there be a look-up tabel $\mathcal{L}$ which gives the optimal choices for servers for this initial situation and let $L_i(t) \in \{1 \dots K_i\}$ be the concrete server choice in stage i when there is t time units left until the deadline, according to the look-up table. In the table $L_i(t)$ is defined for $t \in T := [0, \delta]$, where δ is the final deadline, except for the first stage, where only $t = \delta$ needs to be considered.

Now suppose we determined the service distribution for $C_{a,b}$ has changed, and let its new PDF and CDF be denoted by $f'_{a,b}$ and $F'_{a,b}$, respectively. Which values in our look-up table should we adjust? Let $\mathcal{L}'$ denote the new look-up table with values $L'_i(t)$, which is again optimal.

Theorem 3.1. *In the setting sketched above, where only the distribution of $C_{a,b}$ has changed:*

i) The change in distribution has no effect on later (higher) stages:

$$L'_i(t) = L_i(t) \ \forall k > a, t \in T.$$

ii) In the stage where the change has happened, if the look-up table is changed, either the old or the new entry is the concrete service that changed[6]*:*

$$\forall t \in T : [L_a(t) = b] \vee [L'_a(t) = b] \vee [L'_a(t) = L_a(t)].$$

iii) The change in distribution may have an effect on earlier stages, even if the stage of the changed distribution doesn't change in the look-up tabel:

$$[L'_a(t) = L_a(t) \ \forall t \in T] \nRightarrow [L'_i(t) = L_i(t) \ \forall t \in T, i < a].$$

Proof. i) Finding the optimal value for $L_i(t)$ is an optimization problem on the value of t, the PDF's/CDF's of concrete services $C_{i,j}$ and the convolution of CDF's of later services, together with $L_k(t)$ for $k > i$. If $i > a$, none of these has changed, so the outcome must be the same.

ii) Suppose for some $t \in T$, $L_a(t) = x$, $L'_a(t) = x'$, while x, x' and b are all different. Let $r(\omega|t)$ be the expected revenue if we arrive at stage a with t time units left and choose ω in the old situation, $r'(\omega|t)$ the same for the new situation. Since niether depends (directly or indirectly) on the changed distribution of $C_{a,b}$, $r(\omega|t) = r'(\omega|t)$ for $\omega \in \{x, x'\}$. This contradicts x being chosen in the old situation and x' in the new situation.

iii) It is enough to give an example: suppose there are two stages, $K_1 = 2$ and $K_2 = 1$. Let $C_{i,j} \sim U(0, j)$ and $C'_{2,1} \sim U(0, 2)$. Let $c_{1,j} = 3 - j$, $R = V = 10$ and $\delta = 3$. Then $L_1(3) = 2$ while $L'_1(3) = 1$, even though $L_2(t) = L'_2(t) = 1 \ \forall t$.

□

[6] unless both concrete services turn out to be equally good, but we will not consider this case. It will only happen in the case when the servers are equal in both distributions and costs or in soem specific and unlikely cases

In the last part it is not astonishing that the look-up table doesn't change in the 2^{nd} state, as it only has one concrete server. If there were more concrete servers in the second stage, butthey were all worse choices than the first one, even in the new situation, the conclusion would however be the same. If the look-up table contains different values in the a^{th} row for some different t ($L_a(t_1) \neq L_a(t_2)$ for some t_1, t_2), it is likely that if $L_a = L'_a$, $L_i = L'_i$ $\forall i < a$ (so: the change is not important. For instance, if the server initially had a probability of 5% to adopt some value larger than δ, and in the adjusted situation the large value changed to some other number larger than δ, this has no effect on the look-up table).

Let $\mathcal{L}''$ with values $L''_i(t)$ be a look-up table that is similar to $\mathcal{L}$, but updated in the stage where the change has happened ($L''_i = L_i$ $\forall i \neq a$ and $L''_a = L'_a$). Let $R(\mathcal{K})$ be the expected revenue when look-up table $\mathcal{K}$ is used on the new situation. Since $\mathcal{L}'$ is defined to be the optimal look-up tabel, $R(\mathcal{L}')$ is larger than or equal to either of $R(\mathcal{L})$ and $R(\mathcal{L}'')$. What can we say about the relation between the latter two?

3.3 Adding resources

Suppose there are N servers $S_1 \dots S_N$ with corresponding service times the random variables X_i, which are i.i.d. and one more server S_0 with service time r.v. Y. Y dominates X_1. All service times must be non-negative. Now we must put our job through service in each of the N servers $S_1 \dots S_N$ in that order, where service in server (stage) $i+1$ starts when the one in stage i is finished. If this happens before deadline δ we get a reward R, otherwise we get a penalty V. We get the option to add server S_0 to one of the N stages, that is, if we add it to stage i we can choose during actual service in server S_i to abort it and let server S_0 do the work. Of course we must pay some money if we choose to use this privilege. Question: to which stage should we add the extra server as to maximize our reward?

I reformulate the situation above. Instead of adding the extra server to a server of choice, we add it to server S_N to form a stage, and get the choice to put the stage before the 1^{st} stage, after the $(N-1)^{th}$ one or somewhere inbetween. This is equivalent to the description before, since the S_i with $1 \leqslant i \leqslant N$ are i.i.d.

If $\sum_{i=0}^{N-1} X_i \geqslant \delta$, the penalty cannot be avoided, so position of the N^{th} stage is not irrelevant. In the case that $\sum_{i=0}^{N-1} X_i < \delta$ the following applies:

If we choose to put the N^{th} server in the end, we have full information of the actual service times of the other servers. Thus we can make the optimal choice α^* (which is the amount of time we let S_N try to complete the service before we defect from S_N to S_0, or equals ∞ if we choose not to change at all). Other choices for this moment instead of α^* are worse, or at best equally good, in expectation. If we however had decided to put the N^{th} server on some other location, say directly before the i^{th} server ($0 < i < N$) we would have had less information.

Another way to say it: if we choose to put the combined stage in the end, we can choose to ignore all information that has come to us between the i^{th} stage and that one (but we will include information gathered within the stage as it unfolds). Then we can choose an optimal switching time α based on that information, but it won't be better than the optimal choice based on all information up till time N: α^*. However the choice for this α is the same as we would have gotten when the combined stage was put in between stage the i^{th} and the $(i+1)^{th}$ stage. Thus for any possibility of the values of $X_1 \dots X_{N-1}$, the situation where the combined stage is last will not give us a lower expected revenue.

If there is a possibility of positive measure that waiting with the combined stage until the end is strictly better than using it earlier, putting the combined stage last is strictly better in general, since it is never a worse choice.

To compare putting the combined stage before the i^{th} stage and putting in the end. Consider the following strategy for the situation where it is put in the end: pretend the i^{th} to $(N-1)^{th}$ stage have not happened yet but will be executed after this stage (denote this strategy by s_n). Since this is a strategy based on information that is available it is legal, and it is not better than the optimal policy s_n^*. However it will be

exactly equal in result as the optimal policy in the case where we had put the combined stage on the i^{th} position (s_i^*). Let $R(s)$ be the expected reward when strategy s is used. Then:

$$R(s_i^*) = R(s_n) \leqslant R(s_n^*)$$

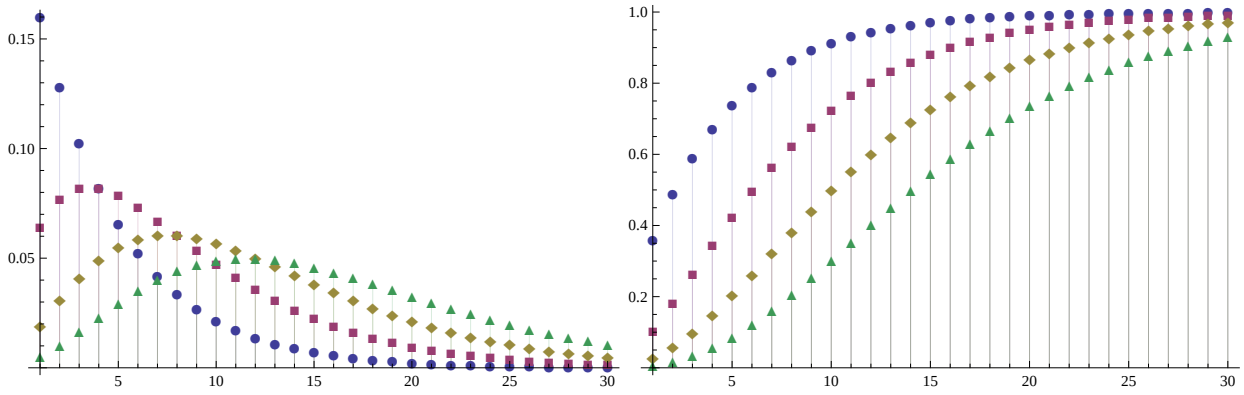
Thus putting the combined stage in the end is the best, or possibly equally good as (some) other locations. For most non-trivial choices[7] of the distributions of X_i and Y however there will be a set of situation in which putting the combined stage in the end is strictly better. If these situations together have a strictly positive probability (which will again be the case) putting the combined stage in the end and then choosing the optimal switching time is also strictly better than putting the combined stage somewhere else.

3.4 Example

In order to gain more experience with and knowledge of the multi-service networks described before, I will now study an instance of such a network, after which I will analyze the effect of some changes.

The system consists of three stages (A, B and C), in each of which there are four servers (1, 2, 3 and 4). Our job has to be processed by a server in each of the stages consecutively. If we finish before the deadline (δ), we get a reward (r). If not, a penalty (v). In each stage, we have to choose one server (based only on the amount of time that is left) and we cannot switch. Each stage has the same set of four servers so we can make a fair comparison between the stages. I have chosen to give the servers discrete distribution for computability. Server 1 will be the most expensive, but also fastest, followed by 2, 3 and 4, which is cheapest, but also slowest.

I chose to use negative binomial service distributions for all servers, because those have a high peak in the beginning and a long tail, and thus resemble service time distributions we have seen in literature to occur frequently in actual service providers. Let X_i denote the distribution of server $i \in \{1, 2, 3, 4\}$, and $X_i \sim NB(i, 0.2)$. This will give our servers the following PDF and CDF:



In the pictures above blue, purple, yellow and green show the values for servers $1 - 4$ respectively. On the x-axis the number of time units is shown, on the y-axis the probability of the distribution haveing a value equal to (PDF) or at most (CDF) that value. Finally I have chosen $r = 20$, $v = 0$ and $\delta = 50$, as those values seemed to give non-trivial (and interesting) results.

[7] Trivial distribution choices may be: such that the deadline cannot be met or cannot be missed, deterministic distributions etcetera)

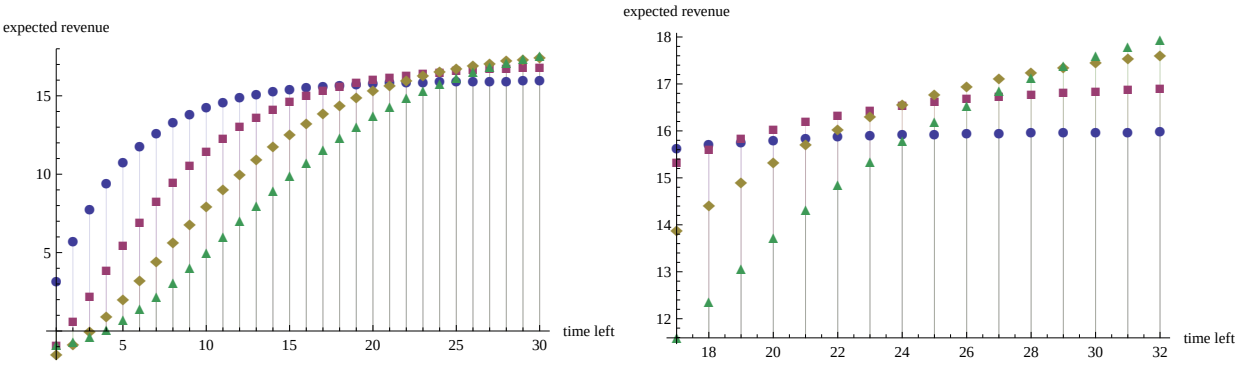


Figure 1: Expected revenue given the time left and the choice for a service provider

3.4.1 Last stage

I start my research at the last stage, as it is the most important one (choices lead to direct profit or loss) and knowledge of it is a prerequisite for the earlier stage.

Let $f_C(t, j)$ be the expected revenue (not taking into account costs in earlier stages) when there are t time units left when stage C is entered and it is chosen to use concrete server j.

In the graphs on the next page, the blue, purple, yellow and green points indicate the values of $f_C(t, j)$ for $j = 1, 2, 3, 4$ respectively. The picture on the right is a close-up of the one on the left.

It follows that concrete service 1 is the best choice for $t \leqslant 18$, concrete service 2 for $19 \leqslant t \leqslant 23$, concrete service 3 for $24 \leqslant t \leqslant 28$ and concrete service 4 for $t \geqslant 29$. So faster and more expensive service providers are relatively a better choice when you don't have time on your side, while slower and cheaper service providers are better used only when time is abundant. At least in the last stage.

What happens if one of the servers is deteriorated? Let $f'C(t, j) = f_C(t - 3, j)$, or equivalently: it is the expectation of revenue when server j is chosen, but it is deteriorated by three time units.

Let $g_C(t) = \max_j\{f_C(t, j)\}$, the maximum expected revenue without the deterioration, and $g'_C(t, 1)$, the maximum expected revenue when only server j deterioration and the others aren't. For small values we would have chosen service provider 1. In the next figure on the left are shown the values of $g_C(t)$ (in blue), $g'_C(t, 1)$ (in purple) and $f'_C(t, 1)$ (in yellow). It shows us that server 1 is still the best choice when there are at

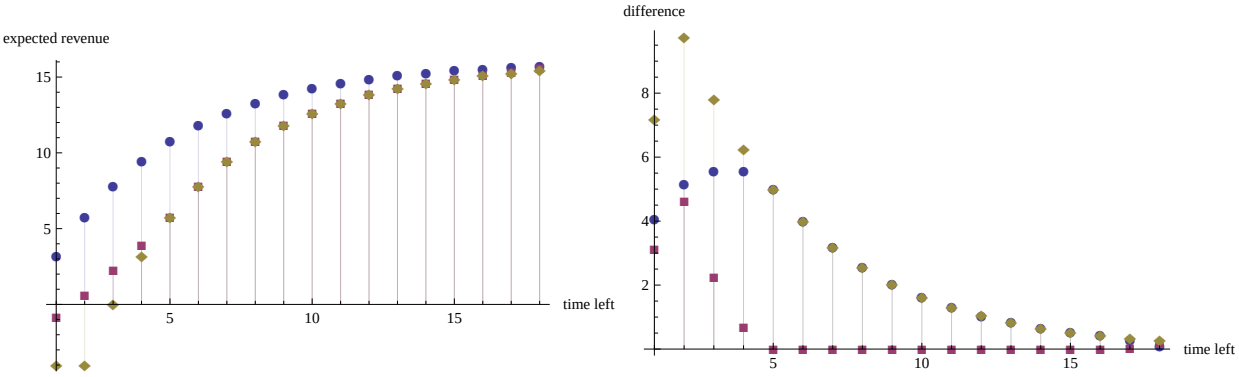


Figure 2: Expected revenue in various situations (left) and their differences (right)

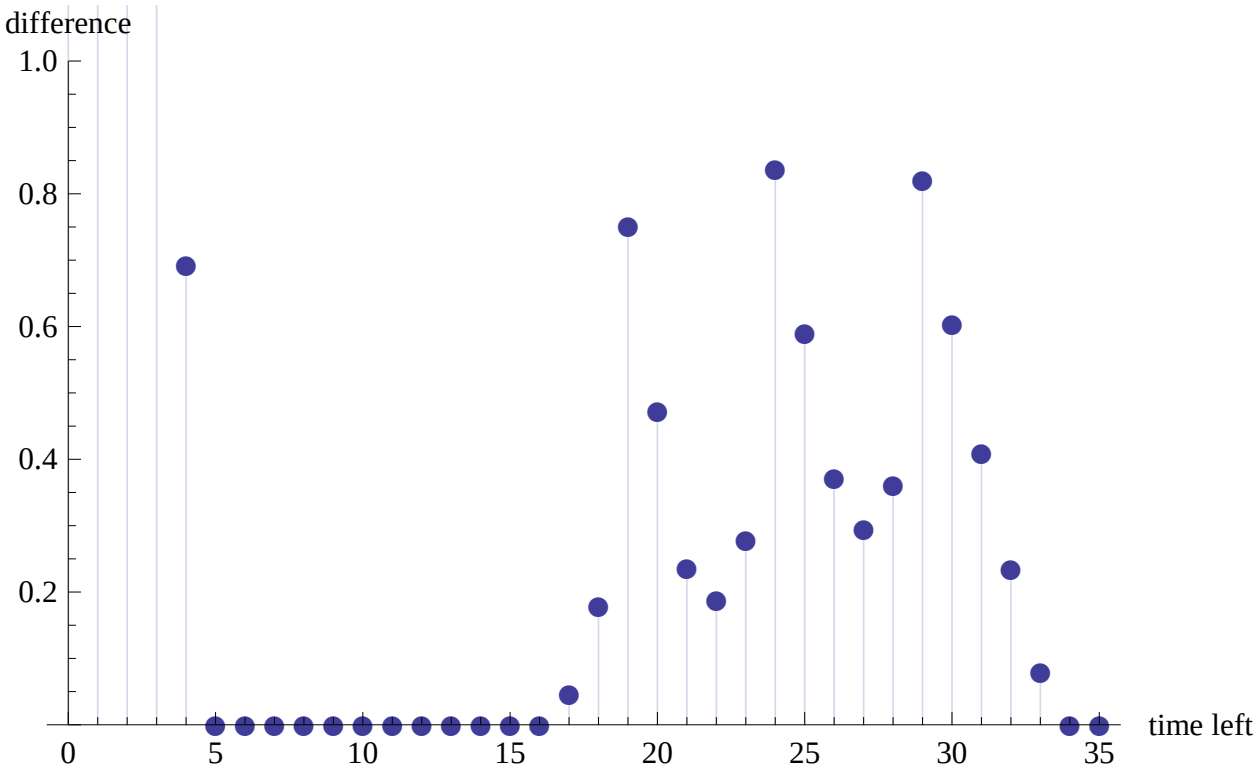


Figure 3: Expected loss when we don't act on deterioration in stage C

least 5 time units left (the yellow and purple point overlap), but also that our expected revenue takes a real hit for small values of t. The picture on the right shows the differences between the various expectations. Here yellow denotes $g_C(t) - f'_C(t,1)$ (the loss due to the deterioration if we don't act on it), blue denotes $g_C(t) - g'_C(t,1)$ (the loss due to the deterioration if we do act on it), and purple denotes $g'_C(t,1) - f'_C(t,1)$'(the difference between acting and not acting).

Finally let $h_C(t) = g'C(t,j) - f'_C(t,j)$, where j is such that $f_C(t,j)$ is maximal for that given t. The graph above shows us the values of $h_4(t)$. It is the difference between acting and not acting on a deterioration of the server that was preferred in the situation without deterioration. The picture shows us that acting is mostly important when two services are almost equal in profitability.

3.4.2 Middle stage

Now let us look at stage B. Again we have to choose a service, but after it finishes we have to go through stage C before we know if we get a reward. This means that if something goes wrong in stage B (such as deterioration) we may be able to make up for it by choosing a faster service provider in the last stage.

In the uppermost and middle pictures on the next page the expected revenues are shown given our choice for server, and the service providers 1-4 are blue, purple, yellow and green respectively. We can see that concrete service 1 is the best if $4 \leqslant t \leqslant 26$, concrete service 2 when $27 \leqslant t \leqslant 31$, concrete server 3 when $32 \leqslant t \leqslant 36$ and concrete service 4 when $t \leqslant 3$ or $t \geqslant 37$. Again faster services are preferable when time is running out, while cheaper are a better choice when time is not an issue yet. One exception is that for very small amounts of time left, the slowest server is preferred! This can be explained as 'giving up'. When only a few time units are left, it is so unlikely that the deadline is met, it is better not to spend any money on

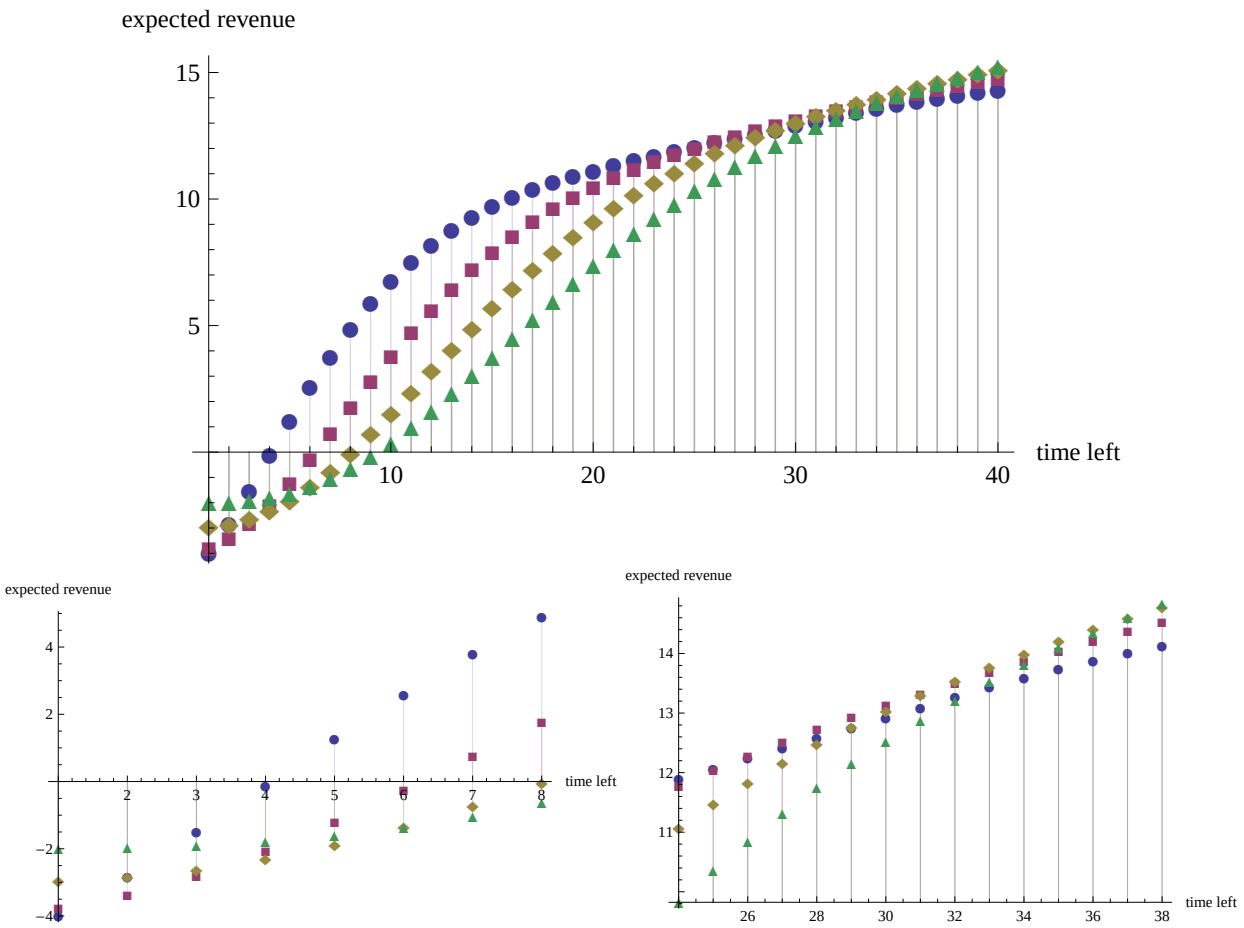


Figure 4: Expected revenue with t time units left in stage B when choosing each of the service providers

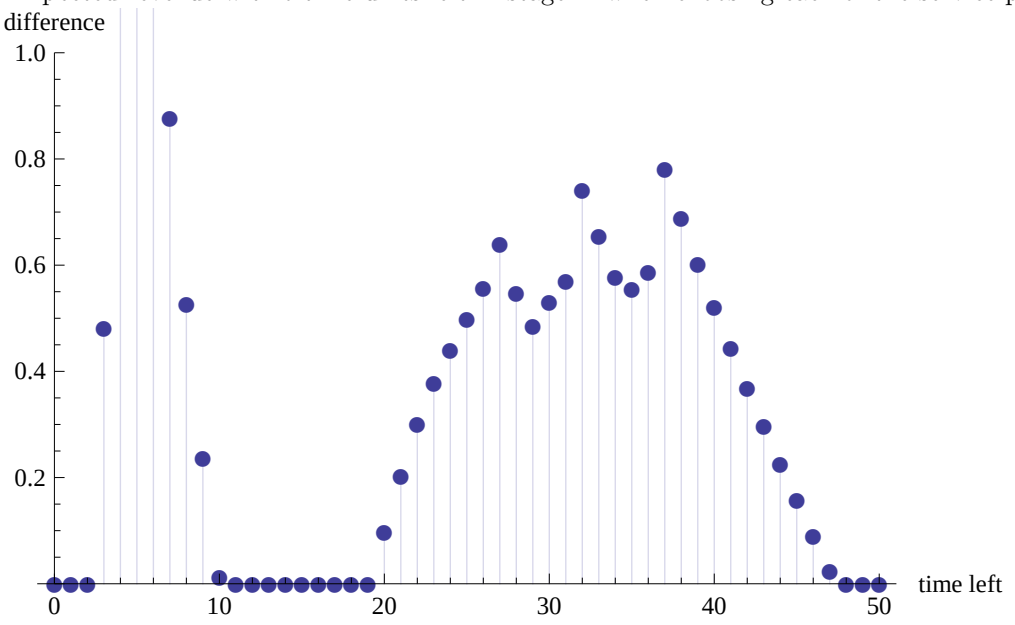


Figure 5: Expected loss when we don't act on deterioration in stage B

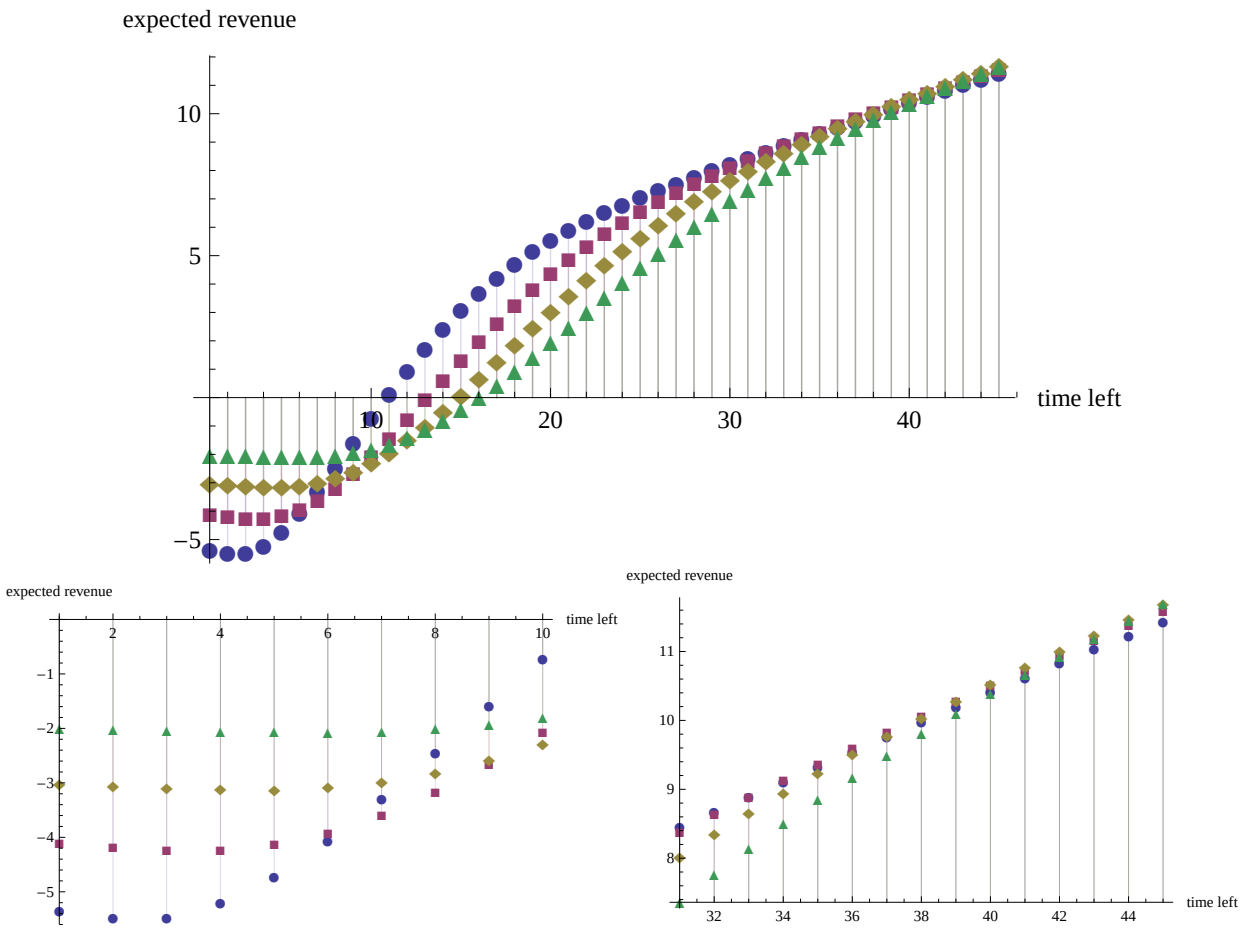


Figure 6: Expected revenue with t time units left in stage A when choosing each of the service providers

fast servers but just take our loss.

Finally let us define $h_B(t)$ (and $h_A(t)$) similarly to $h_C(t)$: as the loss from not acting on deterioration in the optimal server. The lowest picture on the last page shows us $h_B(t)$. It may seem counterintuitive that the loss is larger than in stage C, which was more important. The reason is simple: the choice that is made in stage B is not as important, and thus a small deterioration will already lead to another choice being (much) better.

3.4.3 First stage

Finally we get to the first stage. Actually it would be better to call it the third stage from the end, as it would be easily possible to add more stages before it. We will see however that the results for this stage are similar to the middle stage, and I predict this to be the same for any stages that would be put before this one. The only stage that is really different is the last one. The pictures on this page again show the expected revenues.

Again we see the cheapest server being the best choice when time has almost run out and a preference for servers 1, 2, 3 and 4 on time intervals that in that order. Finally I will show a picture of $h_A(t)$ (yellow), $h_B(t)$ (purple) and $h_C(t)$ (blue) all in one picture. This shows us the amount which we are able to repair

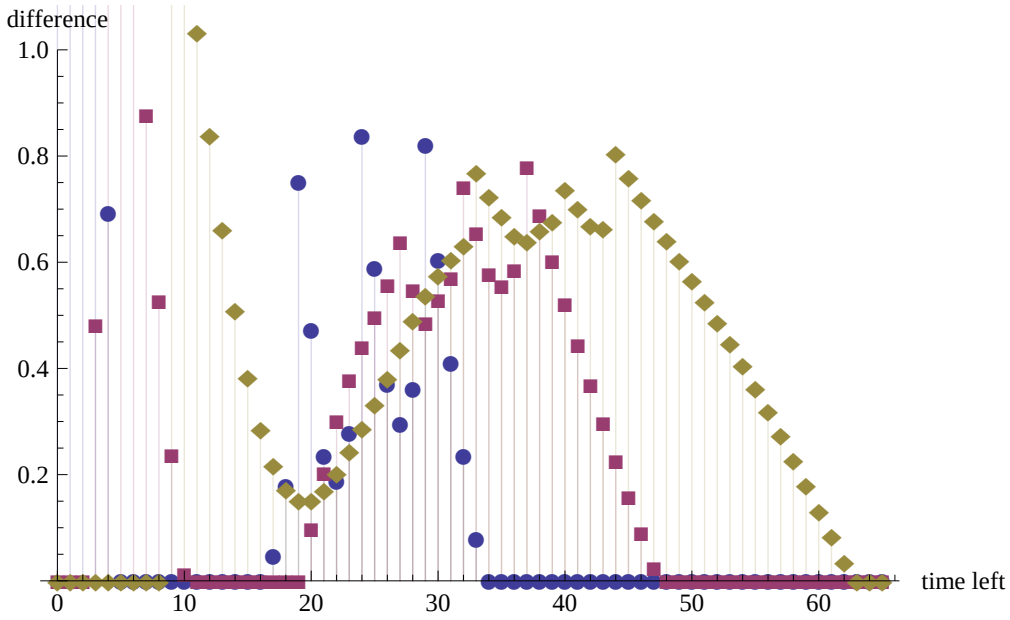


Figure 7: Expected loss when we don't act on deterioration in stage A, B and C

our revenue when our faourite service has deteriorated is similar in each of the stages. Time intervals on which we have to watch for deterioration are:

- When there is little time left and we would have chosen server 1. If it deteriorates, it is better to give up and spend no more money.
- Further to the left, when the service providers have similar expected revenue. This time interval moves to the right when we get to earlier stages.

3.5 Conclusions

In a multi-service network, the most important stage is the last one, because it will immediately affect our revenue if something goes wrong or we make a bad choice. If we have full knowledge of the service distributions, we can find the optimal solution by working back from the last stage to the earlier ones and make a look-up table.

If we are able to add resources to our system, it is best to do it in the final stage, to gain full maneuver space. If we could choose the order of the stages, the latter should have the most options.

It is often profitable to start off with cheaper and slower servers in the earlier stages as to save money, as opposed to later stages. We may need to spend more later to make up for this, but we may also get lucky and don't need to spend that much later. We also have the option to give up, which is profitable if we get unlucky and almost run out of time (although this is mainly due to our Service Level Agreement stating a black-and-white deadline).

Deterioration can have a big effect on our revenue. It is possible to regain revenue from acting on deterioration if we detect it, but only if good alternatives are available. This is more likely to be the case in earlier stages, as effects from choices in those are easier to compensate later. It may be a good idea to act upon deterioration

in an earlier stage than th one it happens in, especially if there are no good alternatives or it is one of the last stages.

4 Internet Experiments

To gain knowledge of internet services time distributions we have performed some experiments. The first set of experiments consisted of measurements of response times of servers we were connected to through the internet. Their setup and results are described in this section.

We will try to find the answer to three questions:

1. What is a typical service time distribution?
2. Do all services have similar distributions?
3. Is the distribution of a given service always the same?

Our hypotheses to these questions are:

1. We expect a large portion of the response times to fall in a small interval. This will be the typical (and also minimal) service time. We also expect a considerable amount of slower responses, and those may be much slower than the typical service time.
2. We are very interested to see other response distributions than the one described above. One distribution we foresee is the bimodal (or multimodal) distribution: one with multiple high peaks and barely anything in between. This could for instance be caused by the service being processed by multiple, unequal servers, or queueing regulations.
3. We expect services to deteriorate when they have a higher workload. Since we have no insight in the workload of the services we use, we will assume workloads are higher during the day than during the night and compare daytime subdistributions with nighttime subdistributions.

The first set of experiments consisted of two seperate series of experiments. First of all we did a pilot experiment to give us an idea what we could expect and draw some first conclusions. For this we sent request from one computer and measured the response times. Later we repeated the experiment with more computers and for a longer time. This is our main experiment. I will first describe the pilot experiment and its results, then the same for the main experiment.

All measured values in these experiments were rounded to milliseconds. Some thoughts on the implications can be found in appendix A.2.

4.1 Pilot experiment

For the pilot experiment requests were sent from a computer on the University of Twente campus to eight servers throughout Europe: one in Catalonia (MURO), two in Munich (LMU and UBM) and one in each of Poznan, Zürich, London, Bremen and Twente itself (different computer than the data requesting computer. The observations in this experiment were uniformly spread over one day, with a measurement done every 10 minutes. This gave us eight time series of ± 150 observations each. The end-to-end response times were rounded to milliseconds. For each of the eight servers I combined the observed values to a distribution. The pdf's and cdf's of these distributions are shown on the following two pages.

In general these distributions have the following shape (qualitative description):

- Most of the probability mass (50-80%) can be found in a small (w.r.t. the average) interval
- (Almost) no observations have a smaller value than this 'peak'

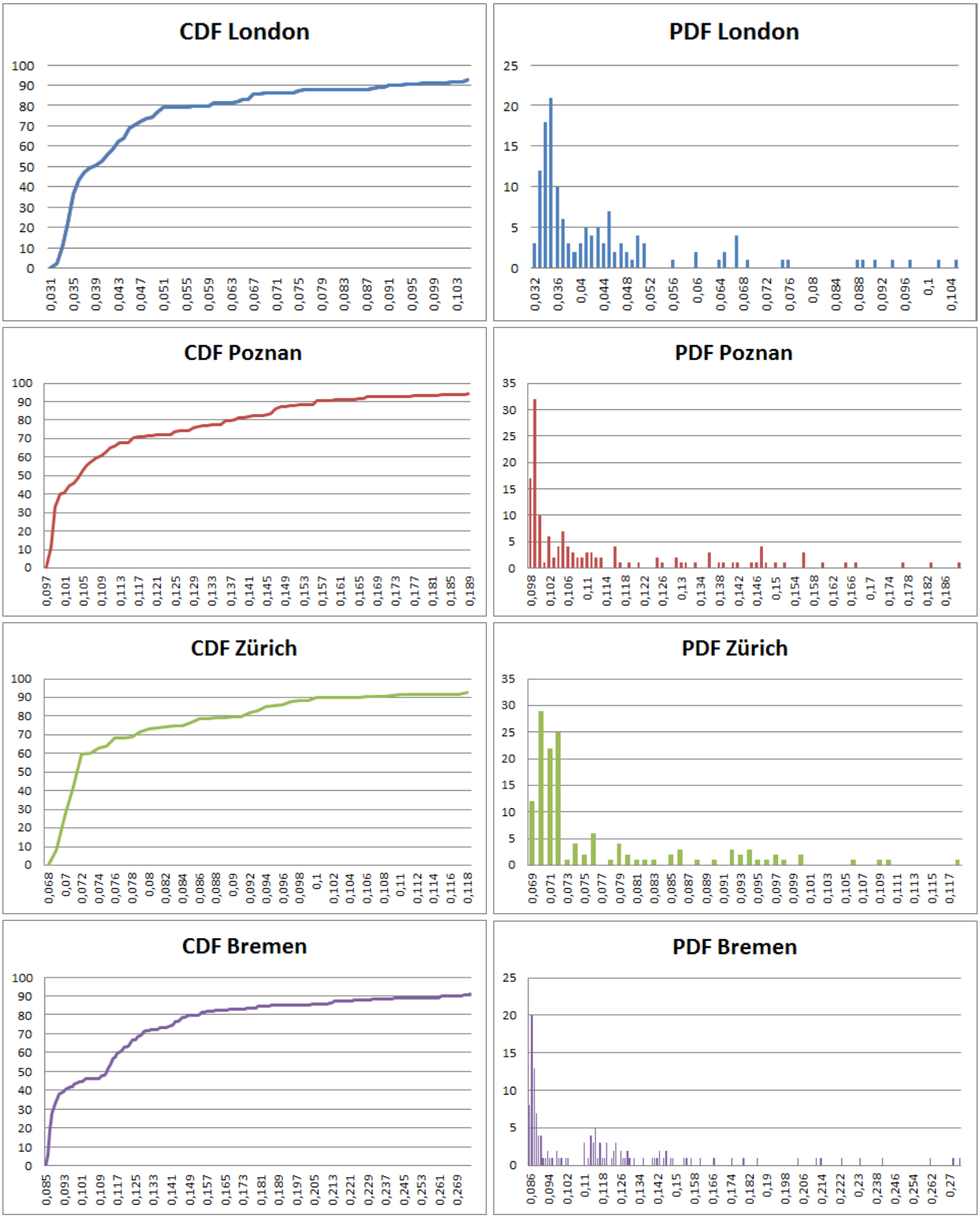


Figure 8: Measured distributions from the pilot experiment

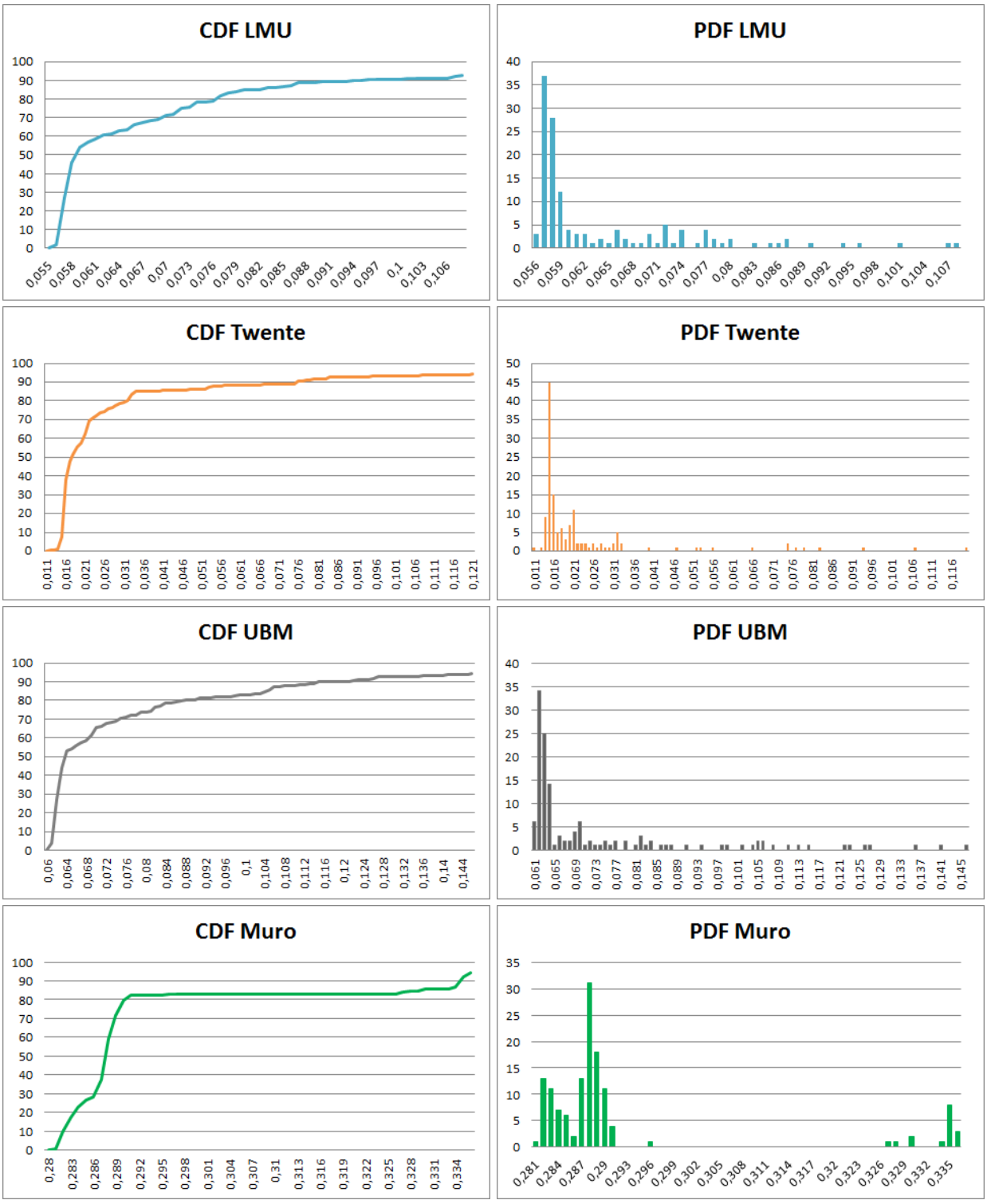


Figure 9: More measured distributions from the pilot experiment

- The distributions have a long (possibly heavy) tail to the right

The first and second bullet points above give rise to the theory that there is a minimum amount of time for the responses due to geographical distance and/or queueing prescriptions at the various servers. When we order the servers by their smallest response time, the ordering we get is similar to an ordening of geographical distances from the server from which the requests were done:

Server	Smallest r.t. (ms)	Distance (km)
Twente	12	0
London	32	480
Munich (LMU)	56	560
Munich (UBM)	61	560
Zürich	69	450
Bremen	86	160
Poznań	98	680
Catalonia	281	1240

The large peak in probability mass then represents the responses that were not delayed much more than required. The small variance within this group of non-delayed repsonses may be due to random variations or measure inaccuracies.

The tail then represents the responses that were delayed. The actual distribution within this tail is very interesting, as unexpected delays in a service or system may cost a lot of money. The number of observations is too small for a thorough investigation however. Some of the distributions have a higher peek than others. The most different is the Bremen distribution, which seems to have a (smaller) second peek 25 ms after the first one. Our hypothesis on general distribution type is strengthened by these results.

4.2 Main experiment

In the second experiment computers in six cities throughout Europe each requested (the same) six http objects. These 36 requests were repeated every 10 minutes. This way we acquired 36 times series of more than 5000 observed response times each.

If we request an http object, our request is sent over the internet to some place where the given website is administered, and there will be one or more computers on that location to handle our request, as well as requests from internet users all over the world.

One measurement works as follows: we instruct one of the servers to request a given http object.

It remembers the time in milliseconds that the request is sent, and waits for the data to arrive, and determines the time in milliseconds when all data of the given website has arrived. The response time is now the difference between these two times. It includes the time it takes for the data to travel back and forth (possibly multiple times) over the internet network, as well as the time it takes the service provider to handle the request (again it may have to do some work multiple times).

We requested http objects from the following six web pages:

1. `www.buienradar.nl/images.asp` (Netherlands)
2. `gratisweerdata.buienradar.nl/radar.php` (Netherlands)
3. `www.weer.nl/weer.html` (Netherlands)
4. `img.meteogroup.com/meteo/automatic-jpg/tt_new/tt_nl_current_365.jpg`
5. `www.wunderground.com/global/stations/06240.html` (United States)

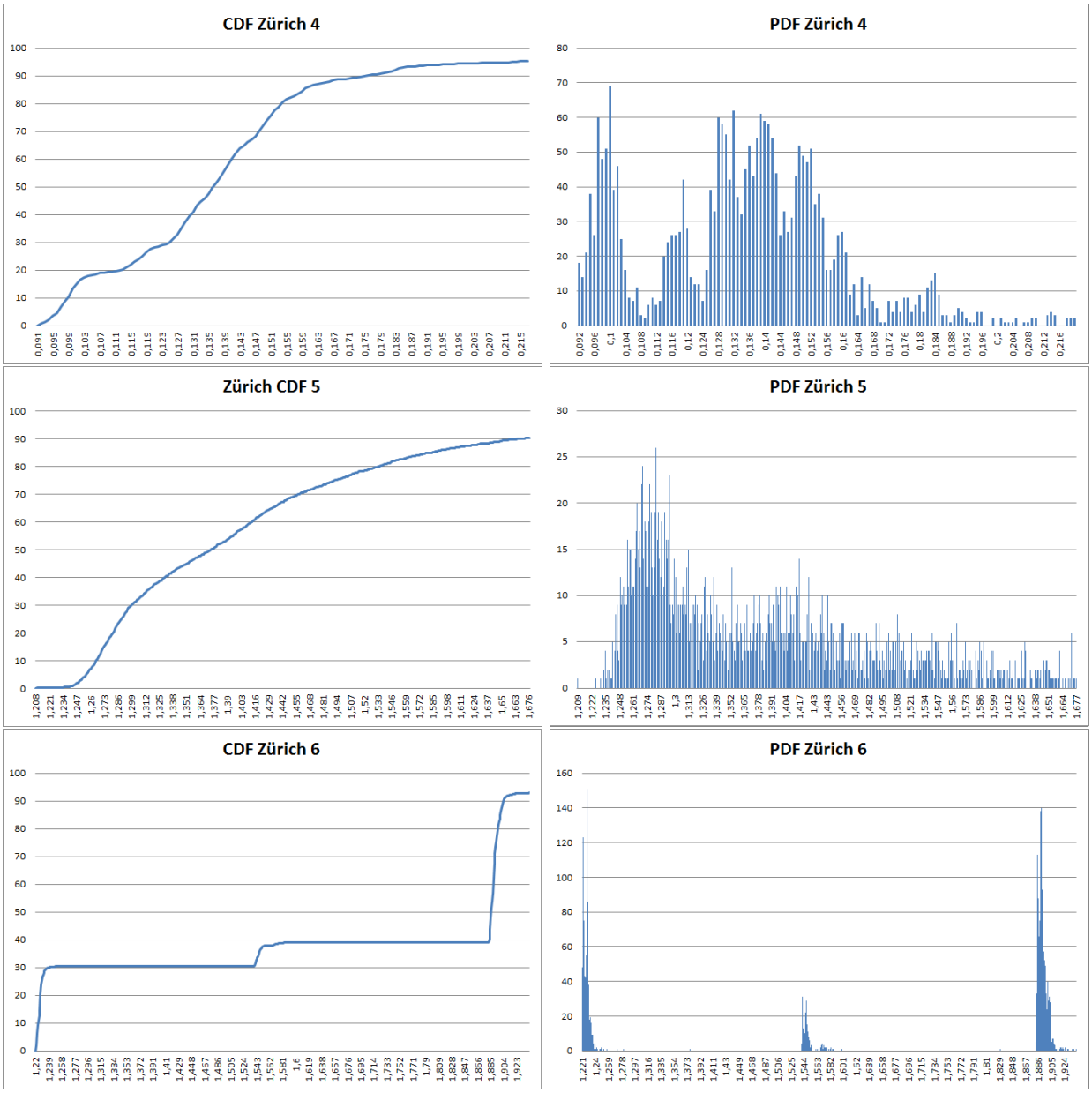


Figure 10: Measured distributions from the main experiment

6. `www.weatherzone.com.au/` (Australia)

Throughout this section they will be identified by their number.

4.2.1 General distribution types

In this subsection I will try to find a general distribution type or possibly multiple types, and see if they match the description in the first two hypotheses.

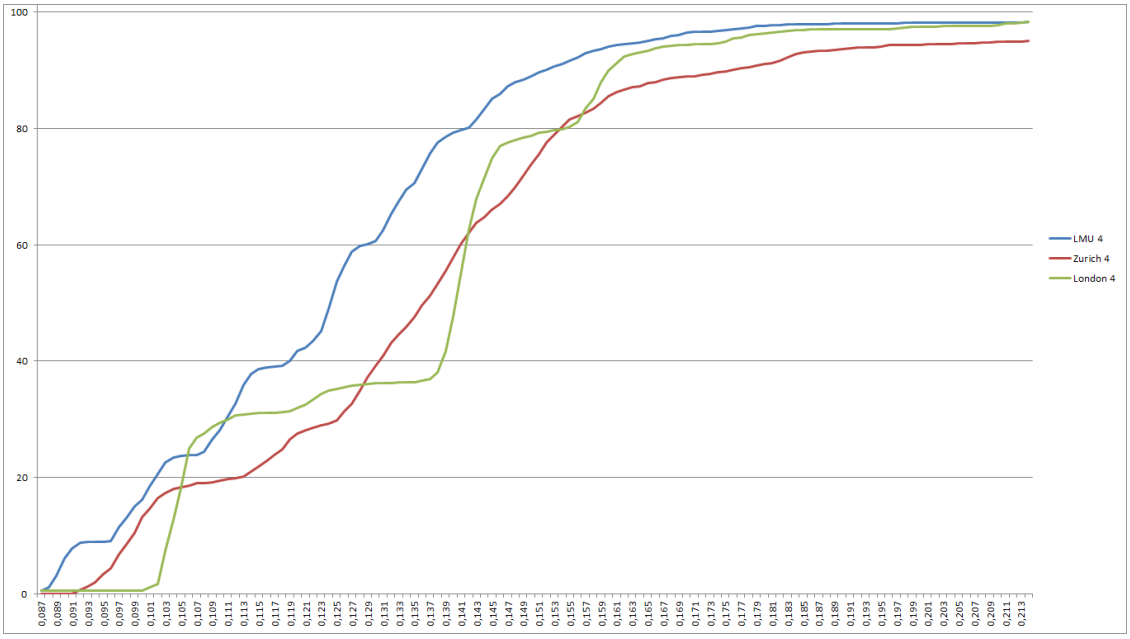
I chose Zürich to be my first subject of study. On the following page the measured PDFs and CDFs of the response times from web page 4, 5 and 6.

The distributions are kind of different from our pilot experiment: the measured values are more dispersed. It can be explained from a main diffence between the pilot and main experiments. In the first we only 'pinged' the other computer – requested it give a notice of 'message received', while in the latter some data had to be sent, consisting of possibly multiple packages, each of which could get delayed or even lost. Still the measures times are not really far apart. Again we see a hard minimum with no measured values below it.

While the response times from websites 4 and 5 show a pattern similar to each other and also to our first hypothesis, website 6 is special. It shows a bi- or trimodal distribution as foretold by hypothesis 2.

I looked at a few more ditributions drawn from the use of website 4. Their PDF and CDF are on the next page.

The distribution from London showed a multimodal distribution. There were two larger peaks and two (or even more) smaller ones. The one measured from Munich (LMU) also showed a multimodal distribution: although the peaks are quite close to each other, the small number of measured values inbetween them. Interesting is the small group of measurements which is much smaller than the rest. Finally Moscow also showed a multimodal distribution. Below is a comparison between three of the CDFs.



Comparison between three distributions from the main experiment

All four distributions (including Zürich) drawn from website 4 were multimodal, so we can conclude that it

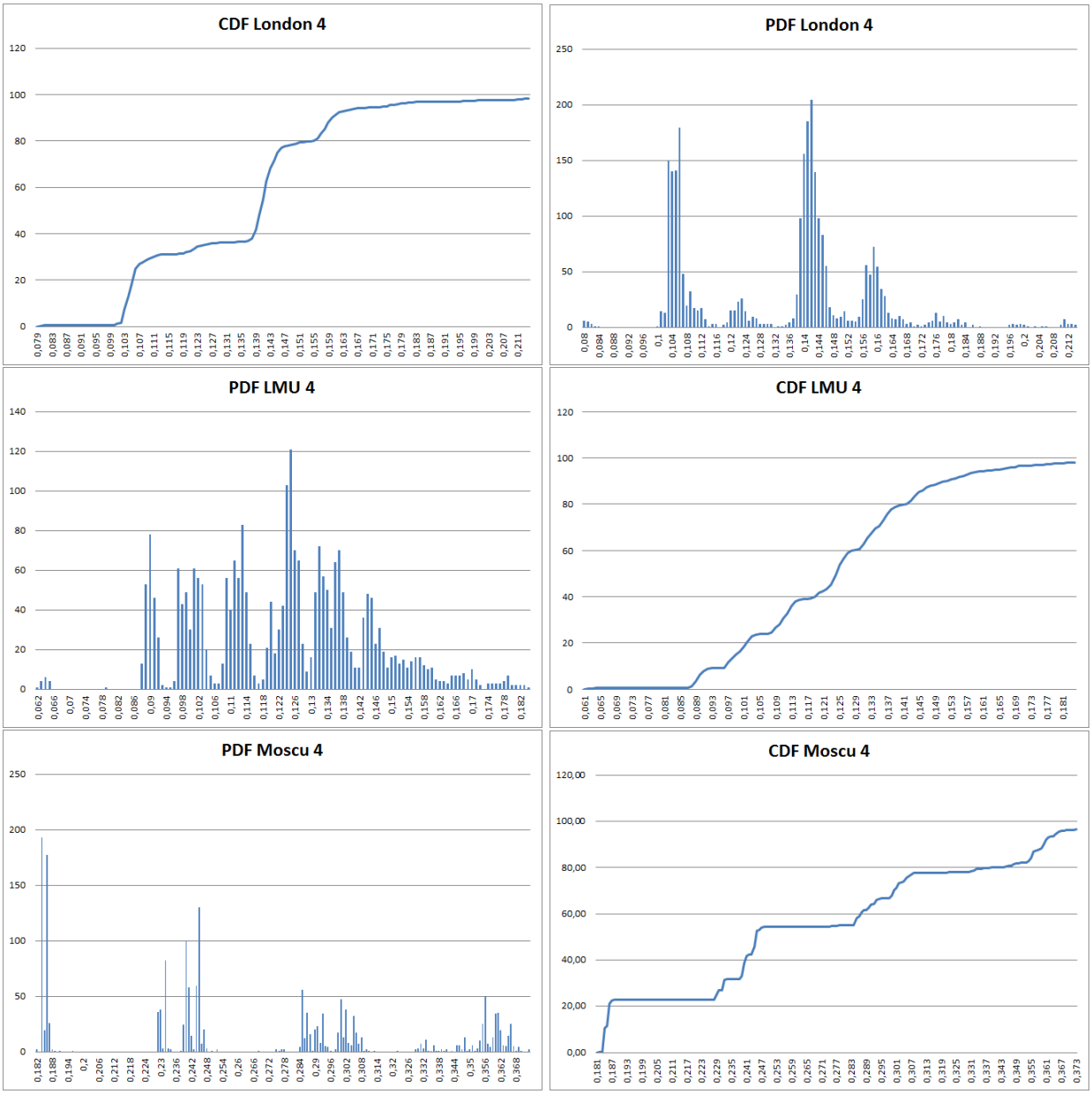


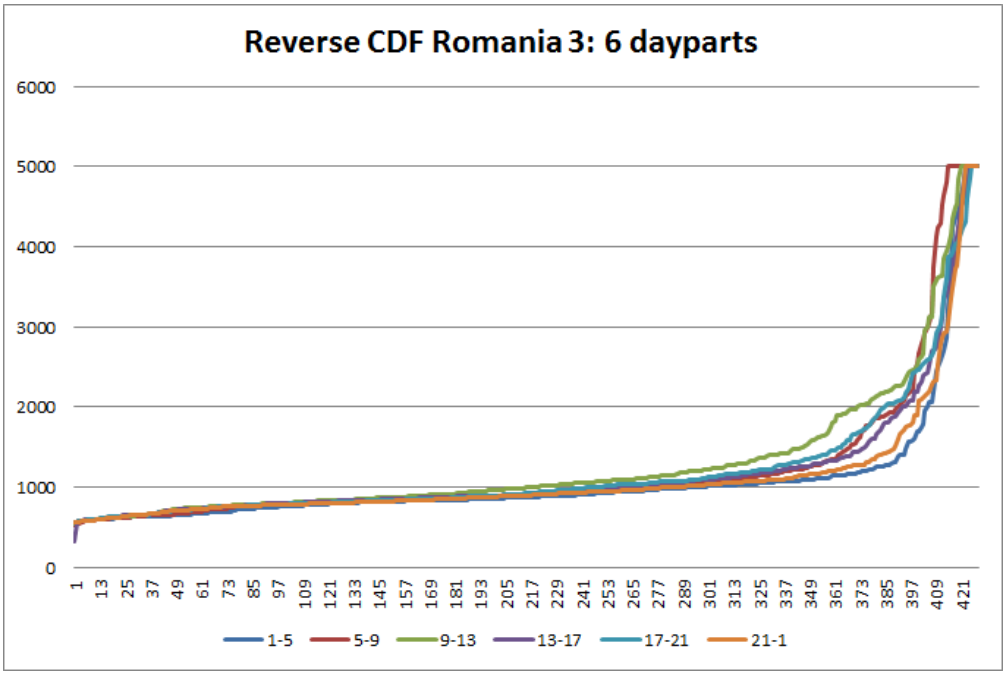
Figure 11: More measured distributions from the main experiment

has a multimodal service distribution. The clear differences between the distributions show the influence of the requesting computer.

4.2.2 Difference between day and night

To test the last hypothesis (that the distribution may depend on load on the server), I drew some subdistributions from the measured sets. I decided to show them using reversed CDFs. This means the values in a subset are ordered from small to large. If the value of a line $n\%$ from the left is now at a certain value x is means that $n\%$ of the measured values in the set the line was based upon are smaller than x. When two of these lines are in the same graph and one is (almost) always higher than the other, it has a distribution with higher values. The reason I did this is averages wouldn't have sufficed, as the outliers have too much impact on those.

To get us started I made a graph for six parts of the day for Romania-3. Measured values over 5000ms are rounded down to that value in the right of the picture. On the x-axis the number of observations is shown.



Reversed CDFs of Romania 3 subdistributions by time of the day

The lines are pretty close to each other, but the one for 9-13 o'clock is a bit above the others, which points towards slower response times at these times, and the ones around midnight are lower. The difference is most visible on the middle to higher end. This leads to the conclusion that any time of the day most of the requests are handled in a regular (short) time, but during the day there is more chance of delay.

The picture for Poznan-3 shows a similar pattern (see next page). The lines for working hours (9-13 and 13-17) are a bit higheer than the ones for evening hours (17-21 and 21-1) and those are in turn higher than those for night and early morning hourd (1-5 and 5-9). The difference between 1-5 and 9-13 is in particular pretty big. Therefore I decided to further look to these periods. I drew a subdistribution for every seperate night of 1-5 and late morning of 9-13. In this picture every nighttime period is red and the morning periods are green (with lightgreen for the ones in the weekend). An average over all observations is also added in black. The observation times are rounded down to 2000 ms if they are larger than this value in both pictures.

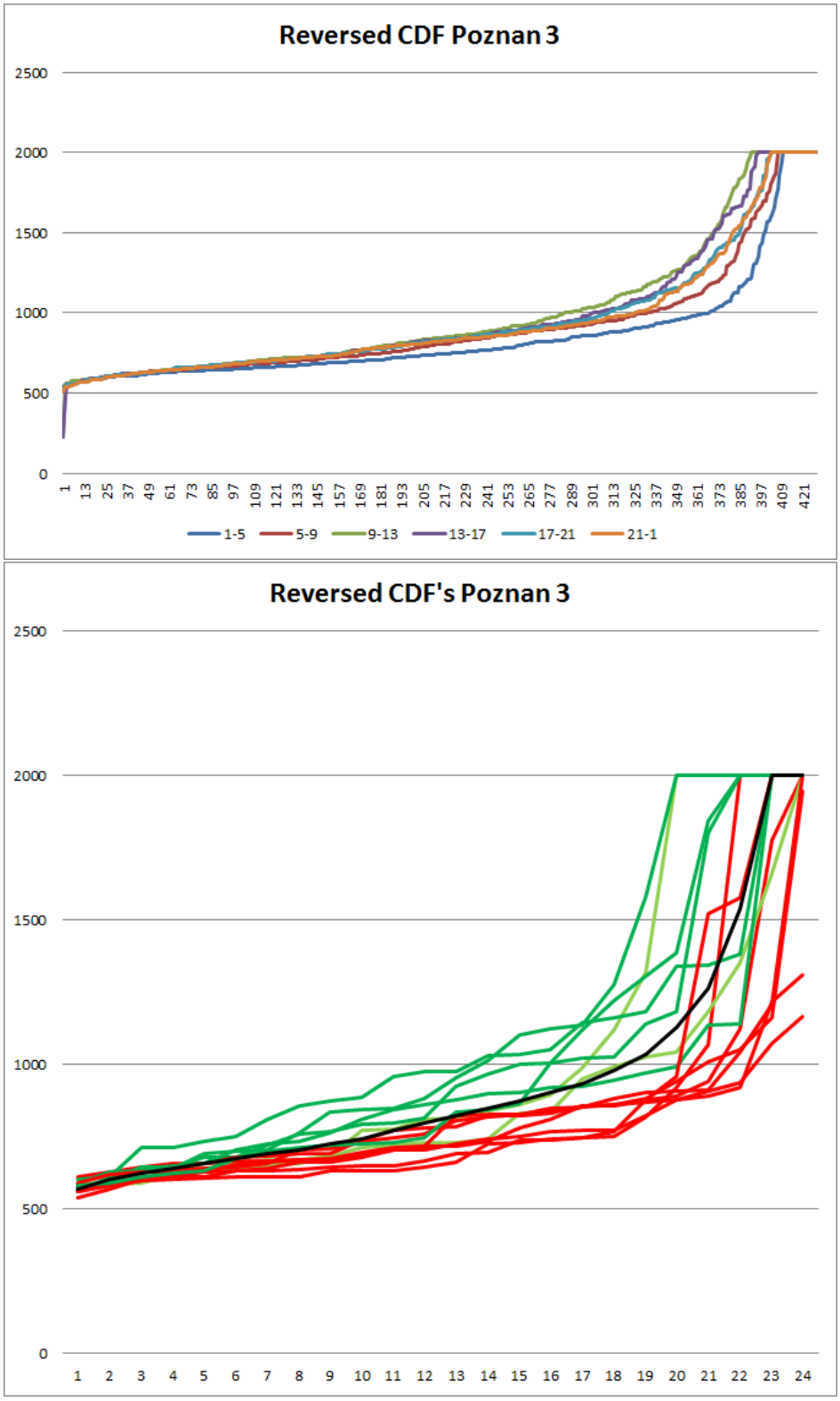


Figure 12: Reversed CDFs of Poznan-3 subdistributions by time of the day

Here weekdays 9-13 are dark green, weekenddays 9-13 are light green and nights 1-5 are red.

I repeated the process for Zürich-5. Since website 5 is American and most likely to be busy during daytime in the (Eastern) U.S.A. (timezone GMT-4). It is thus logical that the most busy times are 13-1 Dutch time (GMT+2), which is 7-19 in the Eastern U.S. I drew another picture with seperate day- and nightperiods, converted to local time. Both pictures are shown on the next page.

4.3 Conclusions

Our first hypothesis, which stated that we expected dense distributions with outliers, I judge to be partly true. The spread of non-outlying values was quite often multimodal, or unimodal but not very dense. The existence of a minimal response time is strongly indicated by the results. The alternative distribution variant stated in the second hypothesis did occur, but a bit more than expected and there were also unclear intermediate variants.

We found strong evidence for the third hypothesis, that service times are slower during the day than during the night. Although the differences in the cases we researched were not enormous, they were clear. This should be taken into account when a model of service time distributions is made. It is not possible to say much about the exact relation between load and speed, as the traffic on the services ae unknown to us.

It also has implications for a multi-service selection prescription. For example it is relatively profitable (fast) to use services during nighttime. If it is possible to choose between servers in Europe and America that deliver the same service, it may be profitable to alternate between them every day and use the one where it's night (assuming it is mostly used by local users apart from us). It is however doubtful whether the effect is larger than the distance factor. Changing preference is more likely to be profitable when a quite large deterioration takes place due to overloading.

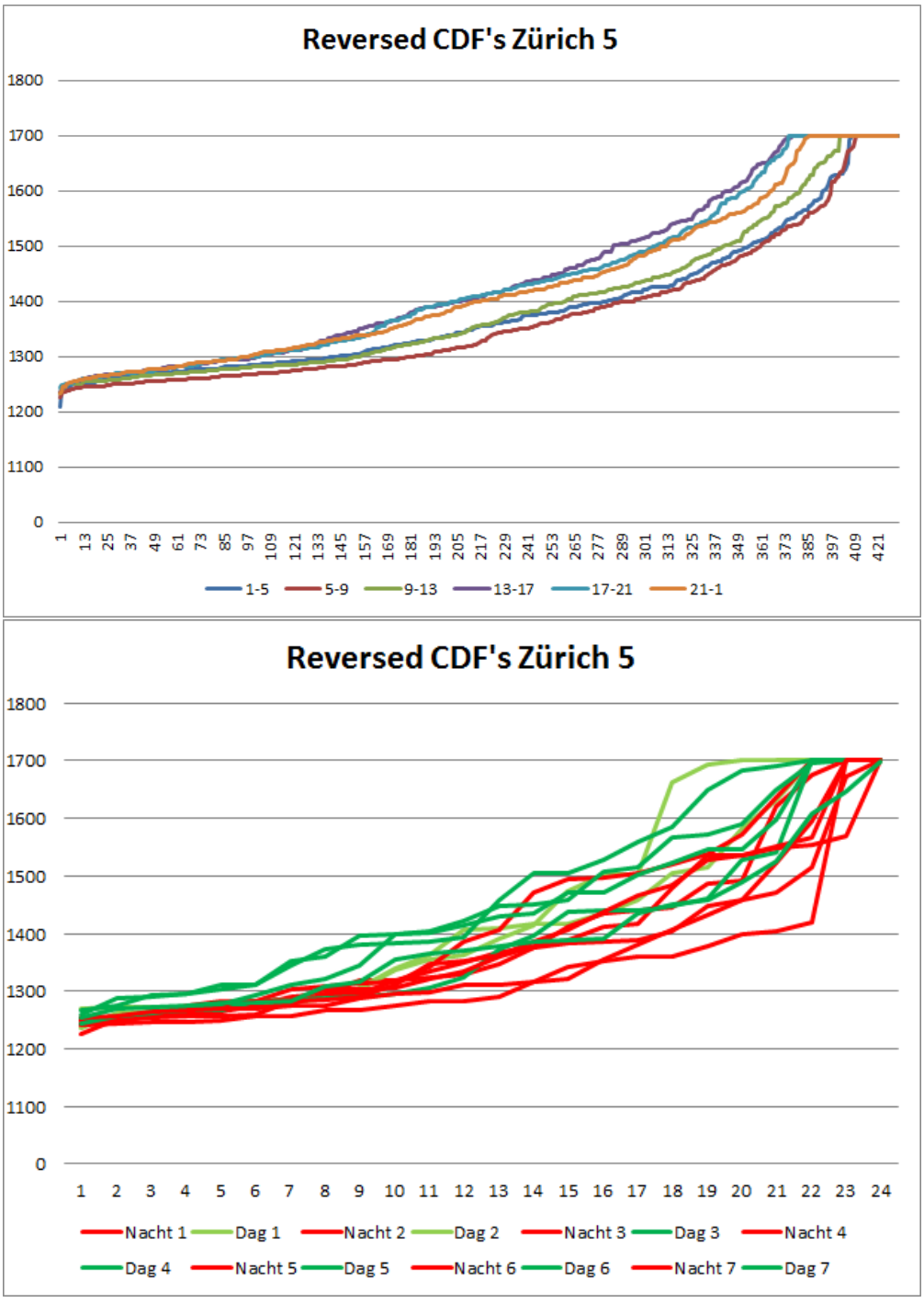


Figure 13: Reversed CDFs of Zürich-5 subdistributions by time of the day

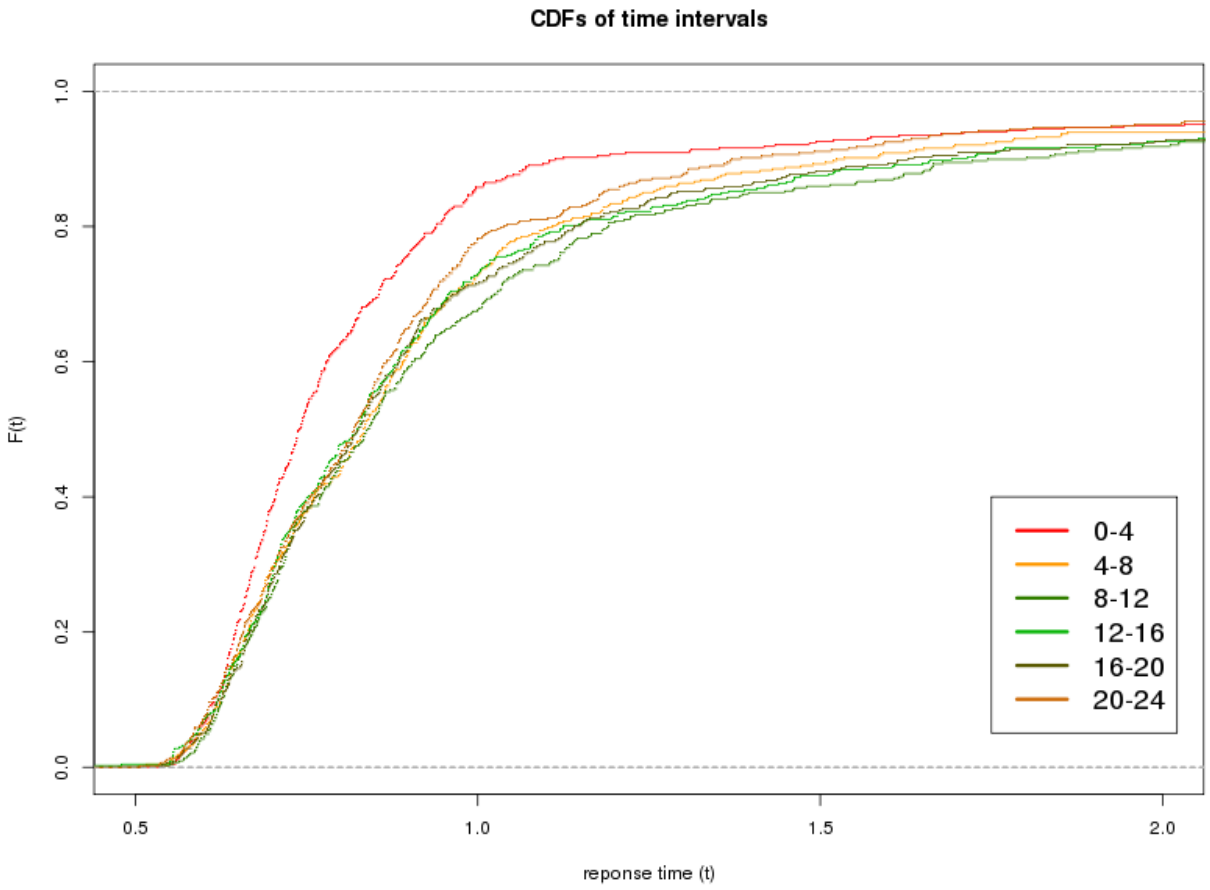


Figure 14: CDF of Poznan-3 subdistributions by time of the day

5 Lab Experiments

In this section I will describe a third series of experiments we performed. This time we did not use computers and services connected through the real internet, but instead did them in a controlled environment ("the lab"), consisting of three linked computers in the same room, which requested data from each other, sent these data packages to each other and measured the response times. First I shall motivate the experiment, describe the setup and summarize the execution. Then I will write about my analysis the data and describe the results. Finally I will comment on the value of the results and draw conclusions from analyses in this section.

5.1 Motivation and Goals

There are two main reason for us to do these lab experiments as an extension to the internet experiments. First of all the actual internet limits our actions. The service providers we used were not controlled by us, and their owners may take action if we would overload their servers with requests. In general, they may monitor our requests and automatically blacklist us when we do many requests in a short time. This could be a problem for the University of Twente, whose computers we used, or for Idilio Drago, who executed the experiments or would give us access tot the computers to do it ourself.

Another advantage of these lab experiments over experiments on the actual internet is that we can fully control the environment. We can increase or decrease background traffic as we like, or change other parameters. This gives us the tools to make the research more fundamental and measure the effect of known changes. A main goal of these experiments is to make a model for internet reponse times.

The biggest disadvantage, which also leads us to think that further research should focus on experiments on the real internet again, is that we can't be sure that our results apply to the actual internet.

The experiments in this section will each have a constant background traffic. For different amounts of background trafic I will try to find out the probability distribution and compare these with each other.

5.2 Setup

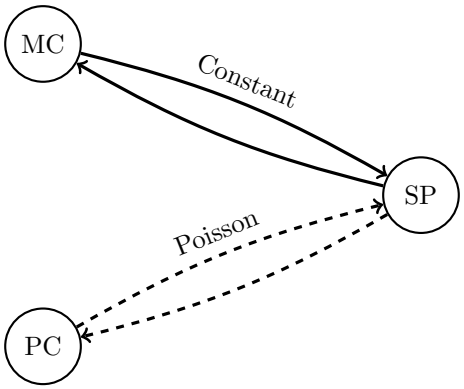
For the experiment we used three linked computers in a closed lab environment at the University of Twente in Enschede. The setup was provided by Idilio Drago. I would use my home computer to log into two of the three computers and command them to send requests to the third. The response times of requests by the first computer would be measured by this computer itself and those make up the data that results from the experiment.

The three computers we used represent a service provider (SP), a customer (measurement computer, MC) and the rest of the internet (parallel computer, PC). The first computer provides two files, a small text file, and a larger xml file. The measurement computer requests the larger file with a constant interrequest times[8], the reponse times usually being bout 1 second and the interrequest times ranging from 1.1 second to 1 minute in the different experiments. The parallel computer requests the smaller file with Poisson interrequest times, ranging from 0 to 500 requests per second. We call the average number of requests per second the "background traffic rate" (BTR).

In the figure, the arrows towards the service provider denote requests and the arrow leaving it the responses. In reality, messages will travel back and forth a few times until all information has been sent. The non-dashed lines denotes that the round trip times between the measurement computer and service provider is being

[8]This is not exactly true, the time between the response and the next request was constant (we will call this number "observation wait" (OW)). However we will see that the variance of reponse times is usually small when compared to its mean value, and thus the interrequest times are close to constant.

Figure 15: Setup



observed and saved. The measured quantity here is the time between the moment the request is sent from the measurement computer and the moment it has detected the requested file is transferred from the service provider. The dashed lines denote that the round trip times between the parallel computer and the service provider are not measured.

5.3 Execution

The experiments in the lab are numbered 1-80. They are performed between October 28 and November 20. The experiments took place in the following manner: first I ordered the parallel computer to start the background traffic at rate BTR, after that the measurement computer with a certain observation wait. After a while I'd stop the measurement computer, followed by the parallel computer. This way, the background traffic rate was constant during the entire experiment.

The parallel computer would send requests with exponential interrequest time, with a rate ranging from 0 to 500 per second. The measurement computer would send requests with a fixed time between the last response and a new request. This fixed time I call the "observation wait" and varied between 0.1 and 299 seconds. As the response times were usually close to 1 second, I sometimes talk of an interrequest time that is 1 second plus the observation wait, although that is not technically true. Also, the request rate is the inverse of the interrequest time, so when the observation wait is x, the request rate is $\frac{1}{1+x}$. When I state that the request rate is y, I actually mean the observation wait is $\frac{1-y}{y}$.

The observation waits used were: 0.1, 4, 9, 29, 59 and 299 seconds, the background traffic took values 0, 10, 20, 30 ... 200, 250, 275, 300 and 500 requests per second. A full list of experiments can be found in appendix A. Not all experiments were succesfull. When the BTR was 300 or higher, the response times would grow fast and after a while the parallel computer would stop working. Some other times the experiment stopped as well, due to either failure in using the right commands by myself or unknown computer errors. I have not used the data from these experiments, and usually repeated them. Some experiments showed unexpected results, but when repeated the new data was always better fitting to our expectations. We assumed the unexpected results were due to errors in these experiments, although it is possible that they were signs that our theories were incomplete.

5.4 Analysis and Results

In this subsection I will describe various experiments and collections of experiments that closely resemble each other. I have only analyzed a small number of the experiments we performed, as there was not enough time to analyze them all. From the result I drew distributions. It is important to realize that the measured PMF (probility mass function, also PDF) is not exactly equal to the underlying PDF. I will regularly comment on the difference.

The measured distributions from the experiments in this section are usually not very smooth and quite irregular. This is caused by insufficient sample size. Hence I have introduced a concept called smoothing, which makes graphs clearer and easier to comprehend. The exact why and how will be explained in appendix A.1. When f is a PMF, I define the n^{th}-order smoothed PMF f'_n as:

$$f'_n(x) = \frac{\sum_{i=-n}^{n}(n+1-|i|)f(i)}{(n+1)^2} \tag{5.1}$$

5.4.1 Low background traffic

For the experiments 1 and 3-7, I used an observation rate of 1/2 per second and background rates 0, 10, 20, 30, 40 and 50 per second respectively. The PMFs are shown in figure 16, its second order smoothing in figure 17. The response times are shown in milliseconds.

As we can see both the mean and the variance increase when the amount of background traffic increases, while the shapes of the PMFs seem to be similar. The following are the means (μ), medians (m), variances (σ^2) and standard deviations (σ) of the observed distributions, all rounded to milliseconds.

Exp.	BTR	μ	m	σ^2	σ
1	0	962	959	702	27
3	10	969	967	582	24
4	20	976	974	426	21
5	30	983	982	582	24
6	40	990	989	923	30
7	50	997	996	159	13

The means are exactly 7 ms apart and the medians 7-8 ms. This gives rise to the theory that, at least for small amounts of background traffic, the average response time grows linearly with the amount of background traffic. This need not be the case for larger amounts of background traffic, as overloading may lead to longer waiting times than expected.

The former pictures imply that the standard deviation should grow when the amount of background traffic grows. However these experiments do not show anything in that direction. This is probably caused by the immense influence of outliers in the variance. For instance in experiment 4, one observation with value 1664 ms was responsible for a 600% increase of the variance. Experiment 1 (relatively) had most outliers that were more than 200 ms above the average, but there were only three, which is still not statistically significant. We should either expand the sizes of the experiments or use a different method to measure variation. I propose two different measures for the variation:

1. Measure the average distance to the median. This will decrease the influence of outliers, as their distances are not squared any more. Let this average distance be denoted by d.[9]

[9] I use the median instead of the mean as the median is the value for which the average distance to it is minimized.

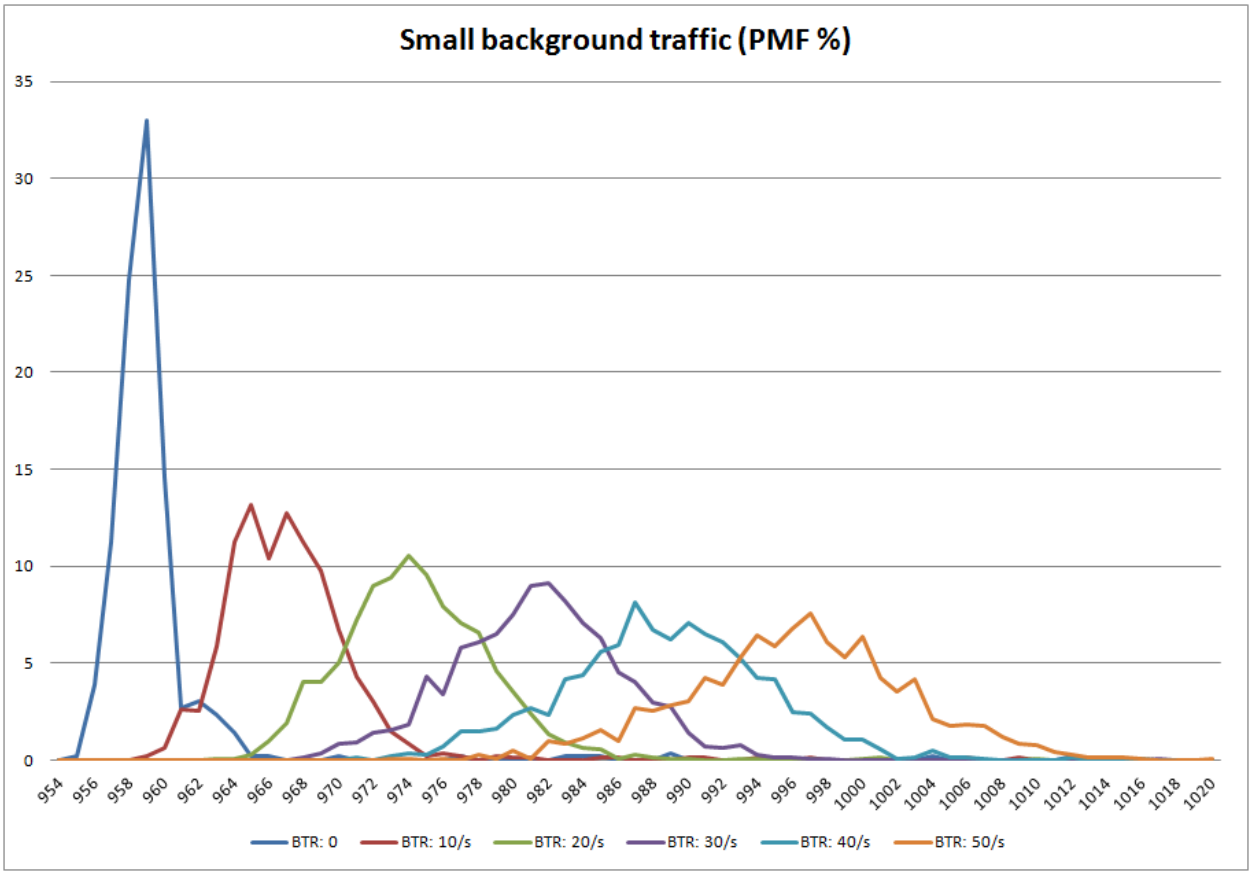


Figure 16: Measured PMFs from experiments with small background traffic

2. Ignore outliers and recalculate mean and variance. I propose to ignore the largest 5% of the values. Let $\mu_{.95}$ and $\sigma^2_{.95}$ be the mean and variance of the remaining observations.

These methods result in the following values:

Exp.	BTR	d	$\mu_{.95}$	$\sigma^2_{.95}$	$\sigma_{.95}$
1	0	3,8	959	1,9	1,4
3	10	4,6	967	7,5	2,7
4	20	4,4	974	13,1	3,6
5	30	5,1	981	19,0	4,4
6	40	6,3	988	27,6	5,3
7	50	5,7	996	34,8	5,9

Now the measures for variation are clearly growing alongside the background traffic. d is still a bit erratic, but again this can be explained from the outliers and the upward trend. $\sigma_{.95}$ and its square clearly have a positive dependence on the amount of background traffic. As can be seen in figure , either of $\sigma_{.95}$ and $\sigma^2_{.95}$ may be linear in the background traffic rate, but it seems the first is a bit concave, while the latter is a convex function of the background traffic rate.

What is the underlying PDF of the PMFs we've measured? The ones we've encountered so far have most of their probability mass on one small interval, and on that interval its shape was similar to a clock, so I

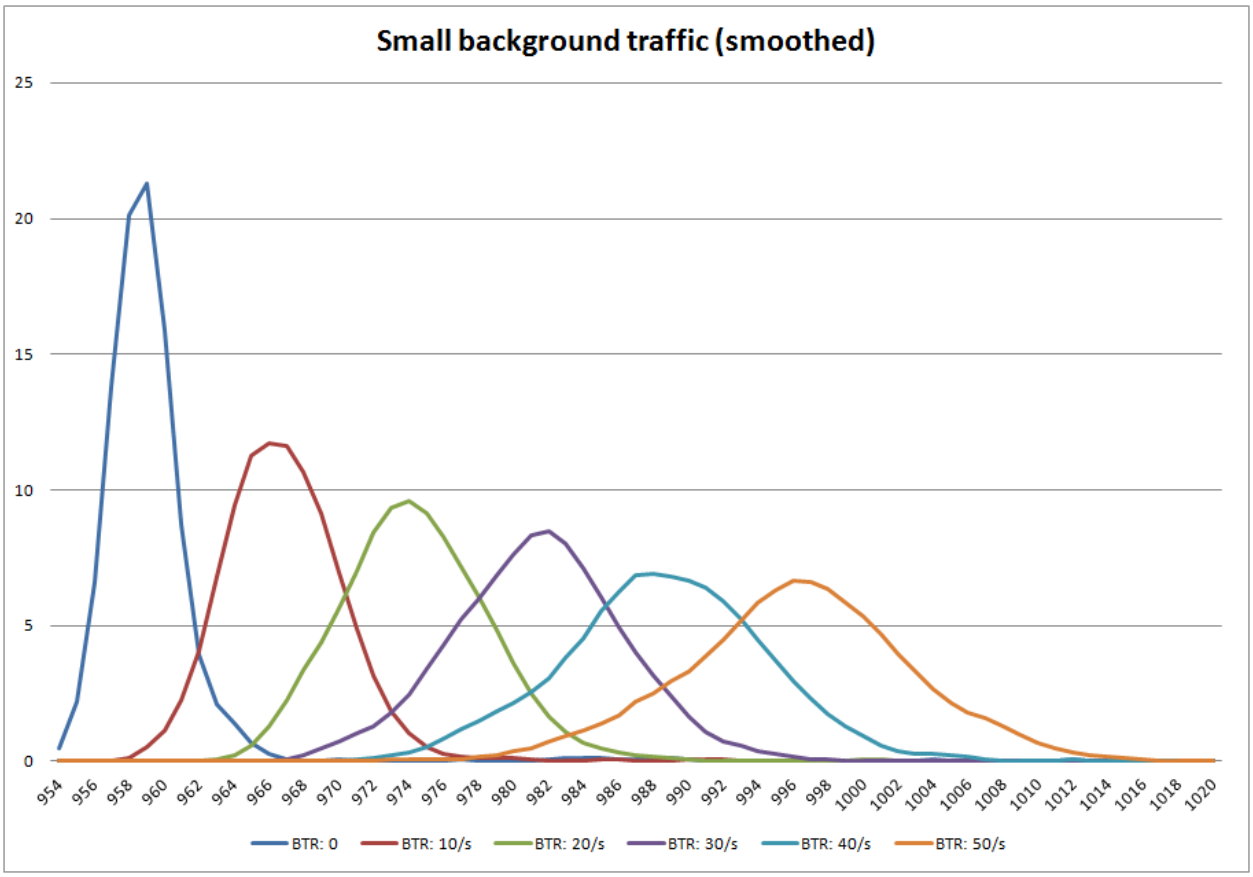


Figure 17: Smoothed measured PMFs from experiments with small background traffic

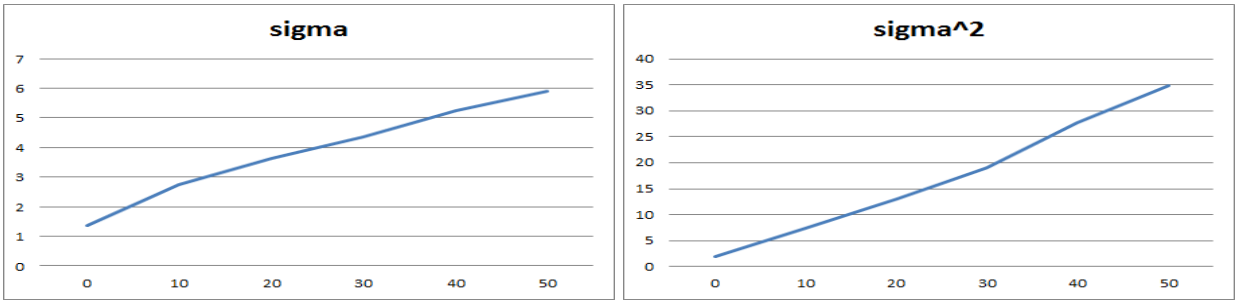


Figure 18: The standard deviation and variance of the measured PMFs (without outliers)

will try to fit a normal distributions to them. Figure 19 shows the observed PMFs from experiments 1 and 3-7, their second-order smoothed version and the fitted PDFs. The following table gives the mean, standard deviation and variance of the fitted normal distributions:

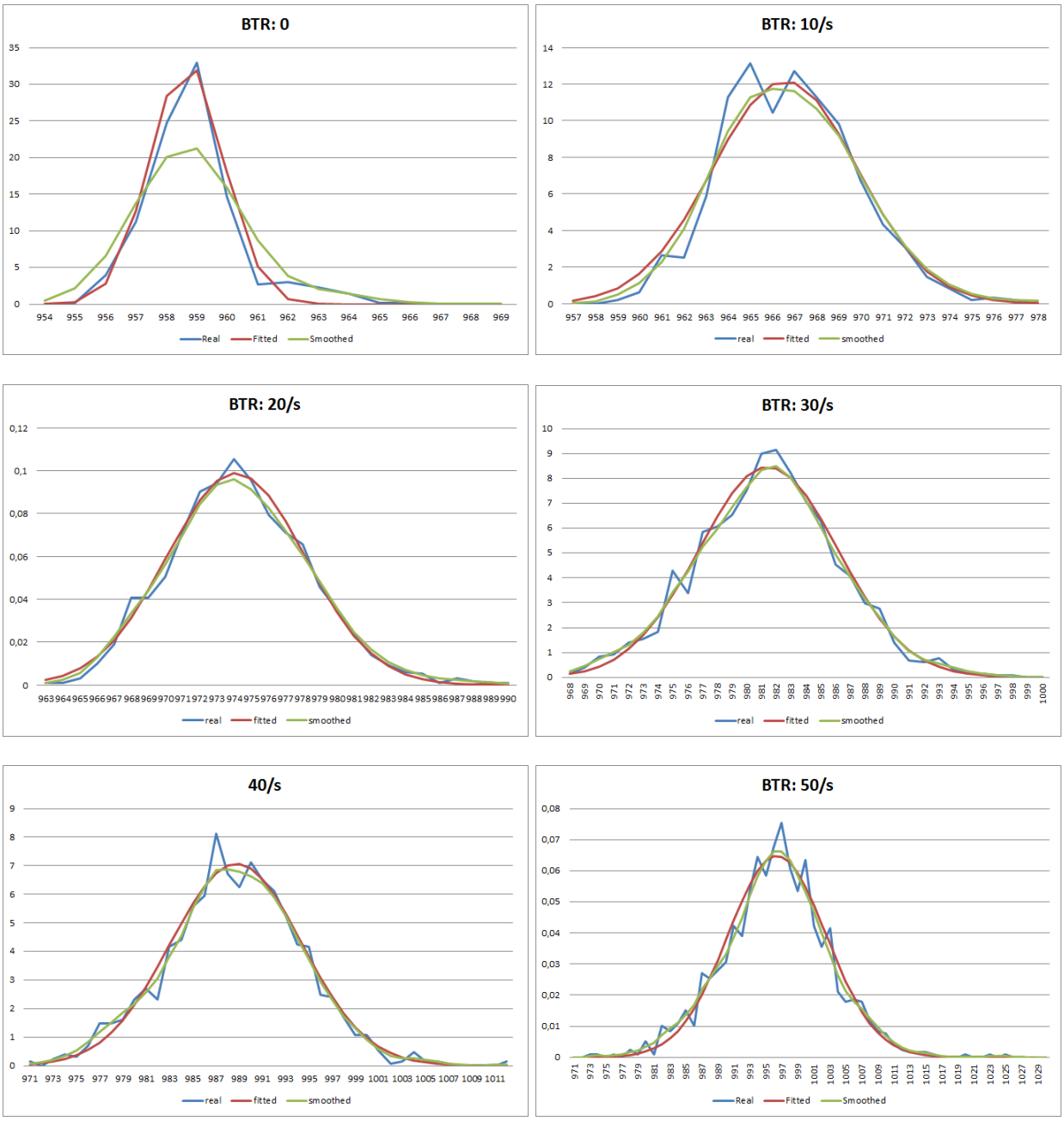


Figure 19: Normal distributions fitted to measured PMFs

Exp.	BTR	μ	σ	σ^2
1	0	958,7	1,20	1,45
3	10	966,5	3,06	9,37
4	20	974,1	4,04	16,31
5	30	981,5	4,71	22,18
6	40	988,7	5,64	31,82
7	50	996,4	6,16	37,90

5.4.2 Medium and high background traffic

I repeated the experiments before with even more background traffic. Experiments 8-11 were set up with 100, 150, 200 and 250 requests per second, respectively. When I tried 300 requests per second or more the setup couldn't handle it and the shut down, which means with these larger values we're getting closer to the maximum capacity. I included the results from experiment 7 (with background traffic rate 50) in this analysis again. These are the PMFs from experiments 7-11 can be seen in figure 20 and the fourth-order smoothed version in figure 21.

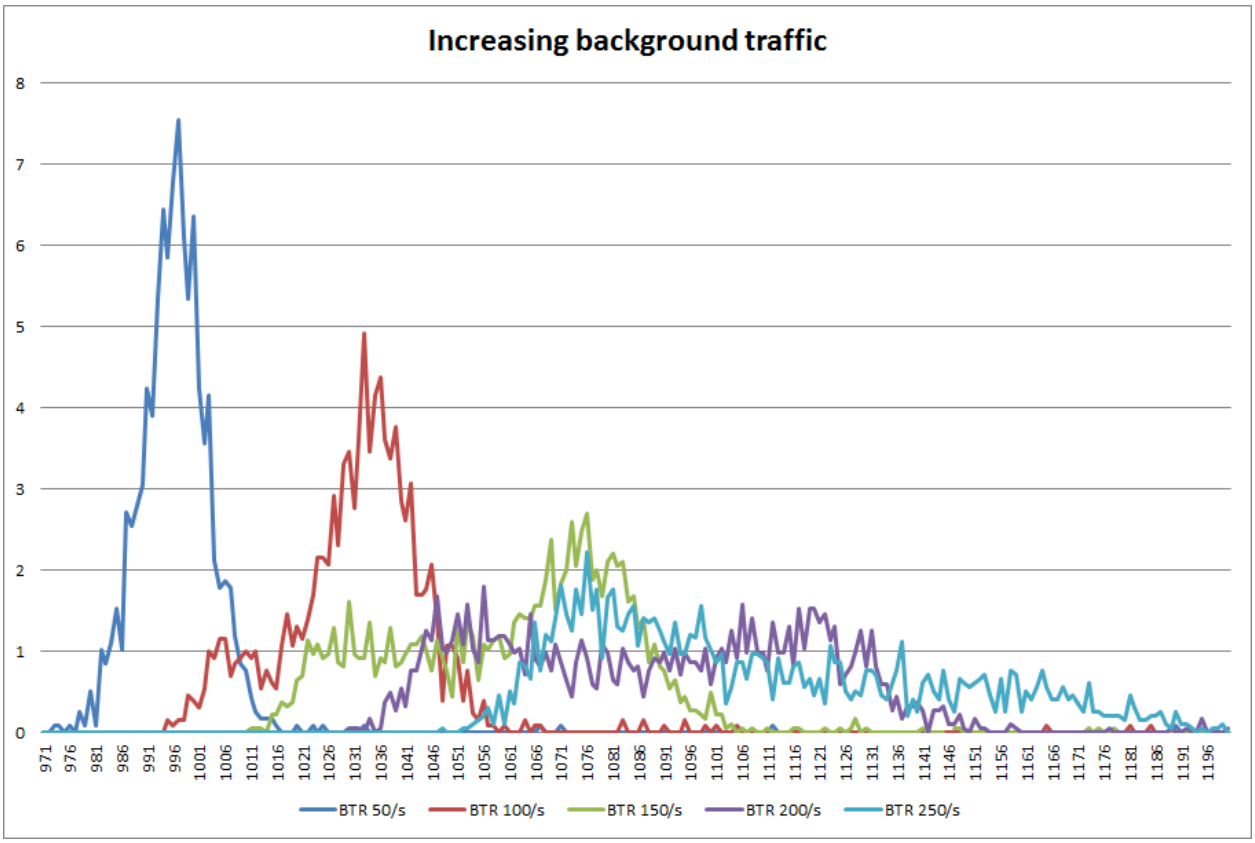


Figure 20: Measured PMFs from experiments with small background traffic

The first two or three PMFs follow the pattern we saw in the experiments with small background traffic, while the others are a bit more messy. They even seem two have two peaks: one where we'd expect it and one at a lower value. The means and medians of experiments 1 and 7 to 11 are summarized below:

Exp.	BTR	μ	m
1	0	962	959
7	50	997	996
8	100	1033	1033
9	150	1063	1068
10	200	1090	1091
11	250	1110	1097

The means start to grow linearly as we saw above, but the growth declines when we get closer to the maximum and the PDFs get their different shape. This decline is counter intuitive, but may be caused

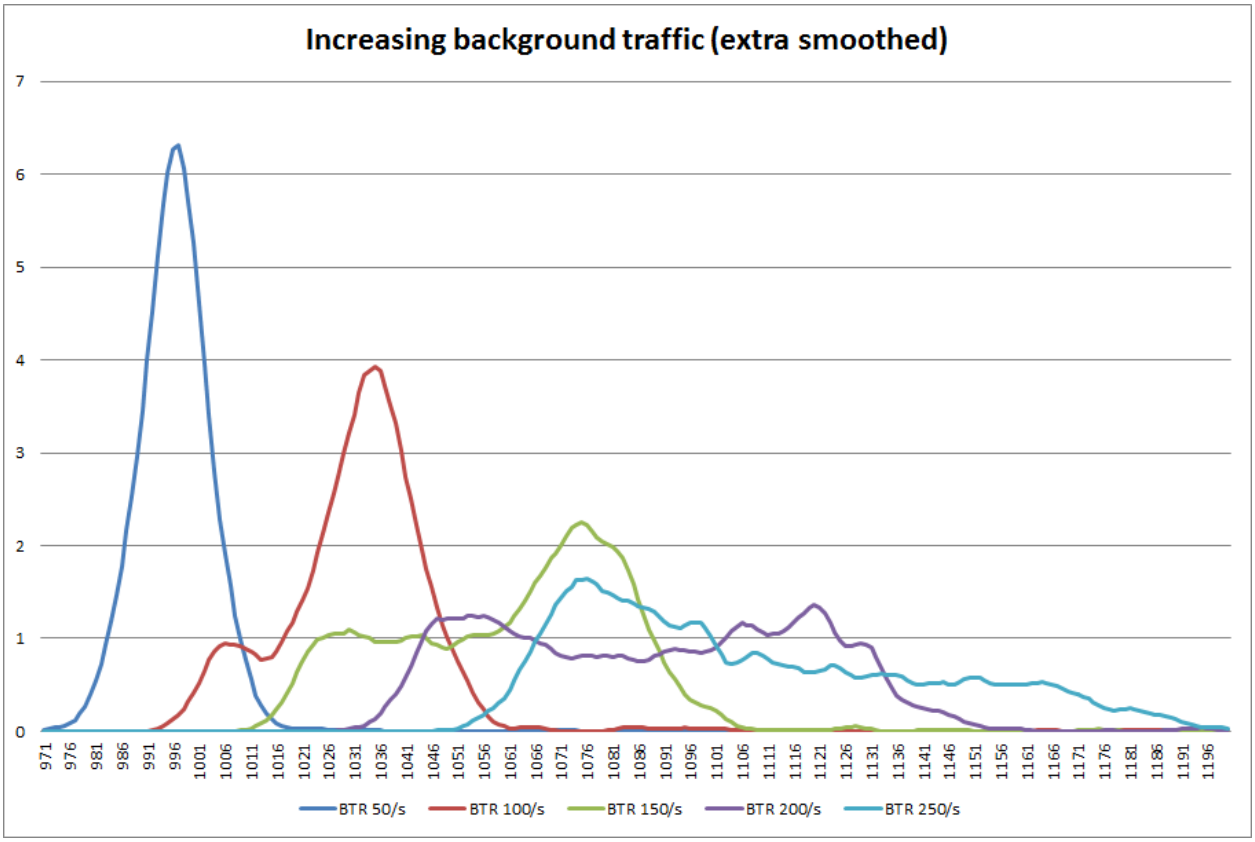


Figure 21: Smoothed measured PMFs from experiments with small background traffic

by problems in the experiment. Maybe the background traffic is lower than the 250 promised in the last experiment, as the cable to the computer is close to its maximum capacity.

Will other distributions also fit well to a normal distribution? For some yes, but the experiments with more background traffic showed different behaviour. We have seen these have PMFs with two hills, so I have tried to fit the one from experiment 9 (with background traffic rate 150) to a double normal distribution (see figure 22).

5.5 Notes on these results

In the 80 experiments together ± 100.000 observations were made. The raw data for each experiment consists of a series of numbers representing the values of the consecutive observations in milliseconds. From this data we also construct a PMF and cdf by taking all observations in an experiment together. I will now try to answer the following questions:

1. How precise are the observations? Is it good enough to measure in milliseconds?
2. Can we trust the observations? If we repeat an experiment, will the results be the same?

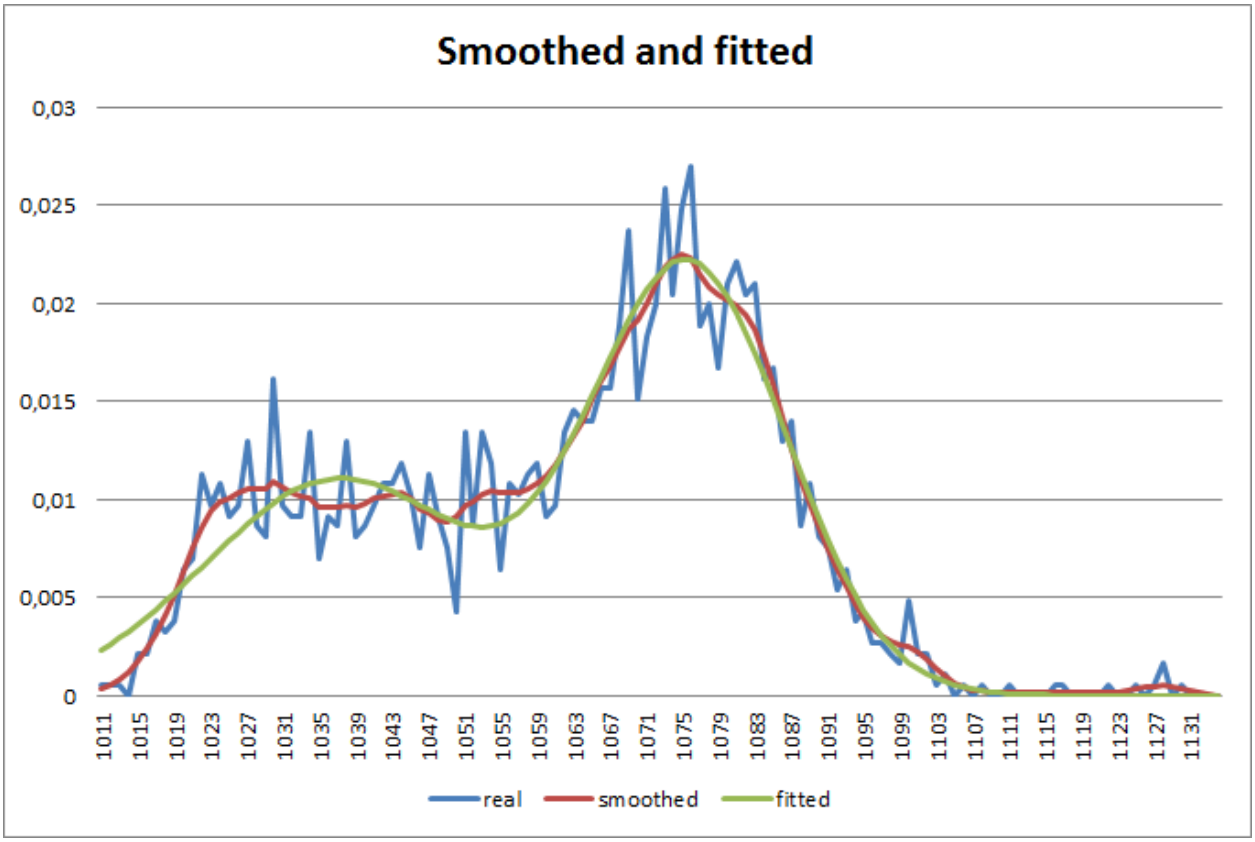


Figure 22: Double normal distributions fitted to measured PMF

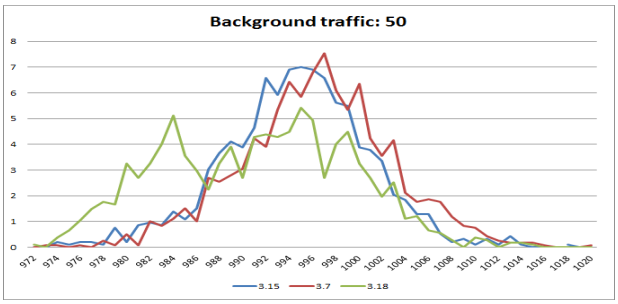
Precision

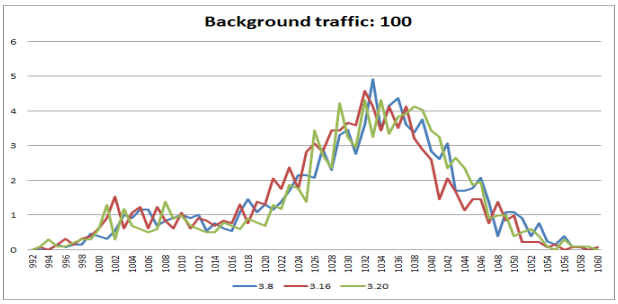
How exact are the results we are getting? SP is not used for other things than these experiments, so that should not be a problem. There may however be a measurement inaccuracy. We cannot promise millisecond precision, but the results are quite unerratic and can be explained from random deviations in the exponentially distributed interarrival times of the background traffic.

In experiment 1 however 84% of the observations were within a range of 4 ms, so even if the actual values were constant, in $\frac{5}{6}$ of the observations the deviation is at most 2 ms. In later experiments variances and deviations are (much) larger, so unless the inaccuracy is also higher for those experiments, the results for those experiments are at least somewhat useful.

Repetition of experiments

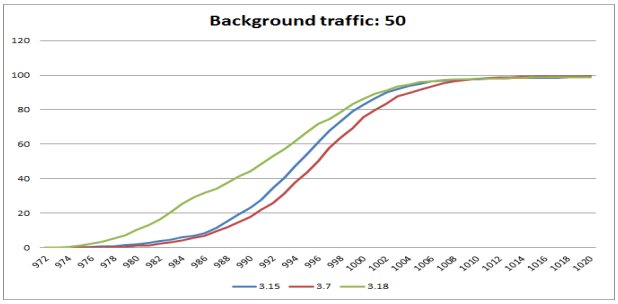
If we repeat an experiment, will the corresponding distributions be the same? They will probably not be exactly equal, but the should be similar.





I have compared some experiments with the same amount of background traffix and the PDFs can be seen in the pictures above. In both pictures the red and blue line correspond to experiments done with an observation wait of 1 second, while the green one is done with an observation wait of 4 seconds. The left picture shows experiments with background traffic 50/s, and the right one experiments with background traffic 100/s.

The picture on the right implies that both repeating an experiment and using a different observations wait have no or little influence on the result (the observation wait should be varied more to confirm that their influence is small). The left picture however shows two experiments that are exactly the same, while the red line is translated a bit to the right. The CDF below shows that the distribution from experiment 7 is about 1 ms larger than that from experiment 18. This would not have changed the conclusions we drew before.



The green line is more to the left, which implies smaller response times. This may be due to the influence of the measurement requests on the total load of the service.

From a certain experiment and onwards, the PMFs drawn from the data seemed a bit of. It turned out the PMFs of these data sets had the same shape as those of earlier data sets, but they were shifted (If X and Y are an old PMF and a new one respectively, then X was very similar to $Y + c$ for some constant c). The shift took place once and overnight, we could not find out the reason.The PMFs shifted to the left with about 10 ms. It worried us at first, but we don't think it is a big problem.

5.6 Conclusions

The main conclusion which can be drawn from the results in this section is that more traffic leads to slower response times. In these experiments the deterioration was almost linear to the amount of background traffic, and this was true for both smaller and larger server loads, up to almost the point where the server went down due to overloading.

Another conclusion is that the variance also grew as a function of the background rate, but the exact growing behaviour is to be determined, but was shown to be substantial.

I would not like to draw conclusions from the negative outliers we have seen, as those are inexplicable and

may very well be due to problems in the experiment setup. It is important to note that the results from this section have not been confirmed with experiments on the real internet. This is recommended as further research.

6 Conclusions

I will repeat the most important conclusions from the substantive sections. The following results were all shown or found evidence for:

- The service for which $c_1 + (r + v)\bar{F}_1(\delta)$ in minimal iss the best choice in a last (or only) stage.
- When in a multi-service network, the last stage is the most important: the choice matters more and there should be the most options to choose from.
- When in a multi-service network, one should typically choose slower service providers in the beginning and faster ones as the stages progress.
- More traffic on a server leads to slower service times. The relation between load and median service time is linear. The variance of the distribution also grows with the load.
- Internet service time distributions are generally either uni- or multimodally dense, usually with a heavy tail.
- Service time distributions can change during the day and service is faster during the night. Thus it could be profitable to alternate between service providers on basis of their timezone.

6.1 Further research

Most importantly the experiments should be repeated and expanded. The lab experiments were very insightful, but need confirmation on the real internet. I would like to see internet experiments where the service providers can be monitored to see their payload and characteristics. Possible expansions include:

- Repeating the multi-service example with different distributions. Possibly distributions which were actually found in internet experiments.
- Quantitatively compare measured distributions using tools such as the log rank test.
- Draw subdistributions by hour from internet experiments. The hypothesis would be that instead of just being slow during the day and fast during the night, service times show a more complex pattern, with highest payload to websites such as the weather websites we chose to research in the morning when people get up, during their lunch break and just before they leave work.
- Take a look at how service times react to changing background traffic, instead of just constant ones like we did.
- Looking for characteristics of overloaded servers: can we predict them going down by their increasing service times or will it always come as a surprise?

Literature

Ivo Adan and Jacques Resing. Queueing theory, 2002.

Sandjai Bhulai and Ger Koole. Stochastic optimization, 2010.

JW Bosman, JL van den Berg, RD van der Mei, HB Meeuwissen, R Núñez-Queija, et al. Dynamic profit optimization of composite web services with slas. In *Global Telecommunications Conference (GLOBECOM 2011), 2011 IEEE*, pages 1–6. IEEE, 2011.

Arnab Chakraborty. The secretary problem – optimal stopping, 1996.

John P Gilbert and Frederick Mosteller. Recognizing the maximum of a sequence. *Journal of the American Statistical Association*, 61(313):35–73, 1966.

Samuel B Graves and Jeffrey L Ringuest. Probabilistic dominance criteria for comparing uncertain alternatives: A tutorial. *Omega*, 37(2):346–357, 2009.

John Rice. *Mathematical statistics and data analysis*. Cengage Learning, 2007.

Sidney Rosario, Albert Benveniste, Stefan Haar, and Claude Jard. Probabilistic qos and soft contracts for transaction-based web services orchestrations. *Services Computing, IEEE Transactions on*, 1(4):187–200, 2008.

Yao Zhang, Zhi-Ping Fan, and Yang Liu. A method based on stochastic dominance degrees for stochastic multiple criteria decision making. *Computers & Industrial Engineering*, 58(4):544–552, 2010.

Miroslav Živković and Hans van den Berg. Analysis of revenue improvements with runtime adaptation of service composition based on conditional request retries. In *Service-Oriented and Cloud Computing*, pages 169–183. Springer, 2012.

Miroslav Živković, Joost W Bosman, Hans van den Berg, Rob van der Mei, Hendrik B Meeuwissen, and Rudesindo Nunez-Queija. Run-time revenue maximization for composite web services with response time commitments. In *Advanced Information Networking and Applications (AINA), 2012 IEEE 26th International Conference on*, pages 589–596. IEEE, 2012.

A Appendices

A.1 Smoothing

Most of the response time PMFs look irregular, having peaks up and down. It is unlikely that they are all caused by the underlying distribution from which they are a sample, but they are probably caused by randomness and insufficient sample size. We can solve this by enlarging the sample size of our experiments, but we don't have time to do that for all experiments. Another solution is smoothing, which means replacing the probability mass functions by a weighted average of the actual PMF and those of some of its neighbours. Denote by f a PMF and by f' a smoothed version. For some values a, b, $(c_i)_{\{-a \leqslant i \leqslant b\}}$:

$$f'(x) = \frac{\sum_{i=-a}^{b} c_i f(x+i)}{\sum_{i=-a}^{b} c_i}.$$

Of course the values c_i should be non-negative. It seems a logical choice to make the smoothing symmetric, so $a = b$ and $c_{-i} = c_i$. Also the PMF of values closer to x should have a larger influence on $f'(x)$ than those further away: $c_0 \geqslant c_1 \geqslant \ldots c_a$.

The PMF's we have seen were sometimes bumpy, as the difference between two consecutive values $f(x-1) - f(x)$ can be considerable. For the smoothed PMF this is not so much the case:

$$\Delta f' := f'(x+1) - f(x) = \frac{\sum_{i=-a}^{a} c_i f(x+i+1)}{\sum_{i=-a}^{a} c_i} - \frac{\sum_{i=-a}^{a} c_i f(x+i)}{\sum_{i=-a}^{a} c_i} =$$

$$\frac{c_a(f(x+a+1) - f(x-a)) + \sum_{i=1-a}^{a} f(x+i)(c_{i-1} - c_i)}{\sum_{i=-a}^{a} c_i}.$$

Let us look at the easiest non-trivial smoothings:

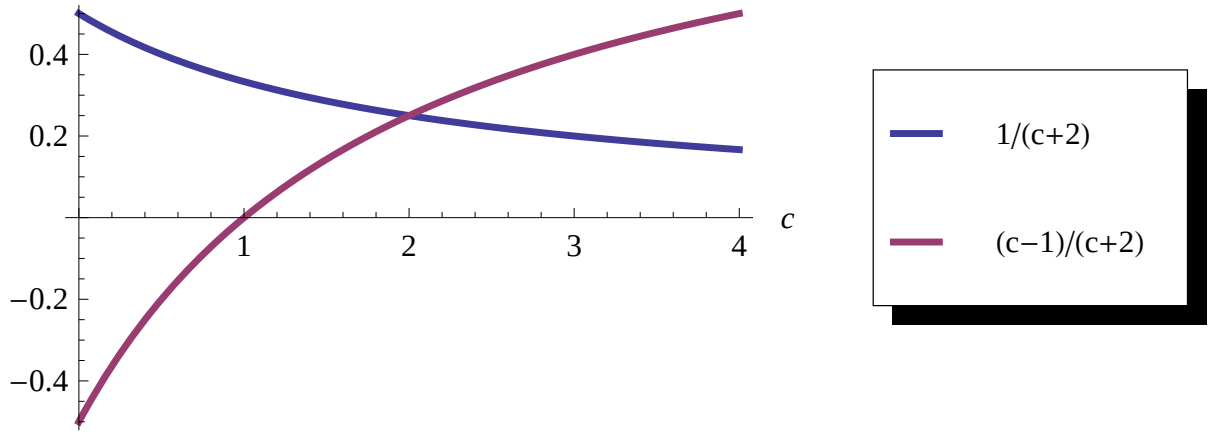
$$f'(x) = \frac{f(x-1) + cf(x) + f(x+1)}{c+2}$$

for some $c \geqslant 1$. What would be a logical value of c? We should consider the effect of c's value on:

$$\Delta f' = \frac{f(x) + cf(x+1) + f(x+2)}{c+2} - \frac{f(x-1) + cf(x) + f(x+1)}{c+2} =$$

$$\frac{1}{c+2} f(x+2) + \frac{c-1}{c+2} f(x+1) - \frac{c-1}{c+2} f(x) - \frac{1}{c+2} f(x-1).$$

If $c = 1$, this difference will be one third of the difference between $f(x+2)$ and $f(x-1)$. This means $f(x+1)$ and $f(x)$ have no effect on the difference between $f'(x+1)$ and $f'(x)$, which seems odd. If $c > 1$, both $f(x+2)$ and $f(x+1)$ will have a positive effect on $f'(x+1) - f'(x)$, while the other two have a negative effect. This seems natural. The following picture shows the values of $\frac{1}{c+2}$ and $\frac{c-1}{c+2}$ as functions of c.



If we want $f(x)$ and $f(x+1)$ to have at least as much effect on the value of $f'(x+1) - f'(x)$, we need $\frac{c-1}{c+2} \geqslant \frac{1}{c+2}$,and thus $c \geqslant 2$. Values of c much larger than 2 will increase the influence of $f(x)$ and $f(x+1)$ and thus the peaks will remain. Therefore I choose to take $c = 2$. Thus the first order smoothing of f I designed is the following:

$$f_1'(x) = \frac{f(x-1) + 2f(x) + f(x+1)}{4}.$$

Now we can extend this method to get higher order smoothings, if the first order smoothing is not smooth enough. Now which values of c_0 and c_1 would be best for the second order smoothing:

$$f_2'(x) = \frac{f(x-2) + c_1 f(x-1) + c_0 f(x) + c_1 f(x+1) + f(x+2)}{c_0 + 2c_1 + 2}?$$

If we first consider integers, a logical choice would be $c_1 = 2$, and $c_0 = 3$ or $c_0 = 4$. The second choice makes the second order smoothing very similar to the first one. Also $c_0 = 3$ will decrease the influence of $f(x)$, so the result will be smoother. I Propose to define f_n as follows:

$$f_n'(x) = \frac{f(x-n) + \ldots nf(x-1) + (n+1)f(x) + nf(x+1) \ldots + f(x+n)}{(n+1)^2}.$$

Or equivalently:

$$f_n'(x) = \frac{\sum_{i=-n}^{n} (n+1-|i|) f(i)}{(n+1)^2} \tag{A.1}$$

For this definition f_n' has the nice property that:

$$\Delta f_n'(x) := f_n'(x+1) - f_n'(x) = \frac{\sum_{i=x+1}^{x+n+1} f(i) - \sum_{i=x-n}^{x} f(i)}{(n+1)^2}.$$

Thus all these values of the PMF have the same influence on the difference between two consecutive values of f_n', while the difference between $\Delta f_n'(x+1)$ and $\Delta f_n'(x)$ is even smaller:

$$\Delta^2 f_n'(x) := \Delta f_n'(x+1) - \Delta f_n'(x) = \frac{f(x+n+2) - 2f(x+1) + f(x-n)}{(n+1)^2}.$$

The last number is usually small (especially if n is large), which means the smoothed funtion has a small second derivative, which will make it look smooth.

A.2 Rounding errors

The observation data is not perfect, as the observations are rounded of on milliseconds. Does this lead to problems? In particular, if (X_i) are the actual values and (X'_i) the data rounded of on milliseconds, does $\frac{1}{n(n-1)}\sum_{i\neq j}|X'_i - X'_j|$ differ from $\frac{1}{n(n-1)}\sum_{i\neq j}|X_i - X_j|$?

Given an observation, the actual response time is within an interval of 1 millisecond. I don't know how the rounding has been done, but this is not a problem. I assume the actual response time is rounded down to the observed multiple of a millisecond. So $X_i = X'_i + \epsilon_i$, with $\epsilon \in [0,1)$. I assume that ϵ_i is uniformly distributed on $[0,1)$. Now if $X'_i \neq X'_j$, without loss of generality $X'_i > X'_j$. Then

$$|X_i - X_j| = |X'_i + \epsilon_i - X'_j - \epsilon_j|.$$

Since $\epsilon_i - \epsilon_j \in (-1,1)$, $X'_i + \epsilon_i - X'_j - \epsilon_j \in (X'_i - X'_j \pm 1)$ and it is thus larger than zero. Therefore

$$|X'_i + \epsilon_i - X'_j - \epsilon_j| = X'_i - X'_j + \epsilon_i - \epsilon_j.$$

Finally given the values of X'_i and X'_j:

$$\mathbb{E}[|X_i - X_j|] = \mathbb{E}[X'_i - X'_j] + \mathbb{E}[\epsilon_i] - \mathbb{E}[\epsilon_j] = \mathbb{E}[|X'_i - X'_j|].$$

However, if $X'_i = X'_j$,

$$\mathbb{E}[|X_i - X_j|] = \mathbb{E}[|X'_i + \epsilon_i - X'_j - \epsilon_j|] = \mathbb{E}[|\epsilon_i - \epsilon_j|] =$$

$$\int_0^1\int_0^1 |x-y|dydx = 2\cdot\int_0^1\int_0^x (x-y)dydx = 2\cdot\int_0^1\left[xy - \frac{1}{2}y^2\right]_{y=0}^{x} dx =$$

$$2\cdot\int_0^1 \frac{1}{2}x^2 dx = 2\cdot\left[\frac{1}{6}x^3\right]_{x=0}^{1} = \frac{1}{3} \neq 0.$$

It follows that:

$$\mathbb{E}[\sum_{i\neq j}|X'_i - X'_j| - \sum_{i\neq j}|X_i - X_j|] = \frac{1}{3}\#\{i,j : X'_i = X'_j\} =$$

$$\frac{1}{3}\sum_z \#\{i,j : X_i = X_j = z\} = \frac{1}{3}\sum_z Y_z(Y_z - 1) =$$

$$\frac{1}{3}\left[\sum_z Y_z^2 - \sum_z Y_z\right] = \frac{1}{3}\sum_z Y_z^2 - \frac{n}{3}.$$

As the Y_z are not larger than n, $\sum_z Y_z^2 \leqslant \sum_z n\cdot Y_z \leqslant n^2$, a maximum which is attained only when all response times are the same (when rounded of to milliseconds). In general the difference will be the largest when there are high peaks in the PDF. Finally:

$$\mathbb{E}[\frac{1}{n(n-1)}\sum_{i\neq j}|X'_i - X'_j| - \frac{1}{n(n-1)}\sum_{i\neq j}|X_i - X_j|] =$$

$$\frac{1}{3n(n-1)}\left[\sum_z Y_z^2 - n\right] = \frac{\sum_z Y_z^2}{3n(n-1)} - \frac{1}{3(n-1)}.$$

Note that if $Y_z \in \{0,1\}\forall z$, so the rounded observations are all different, this quantity is zero, and it is larger in all other cases.

Now how about b? Is the same effect devitating $\frac{1}{n-1}\sum_{i=1}^{n-1}|X'_i - X'_{i+1}|$ from $\frac{1}{n-1}\sum_{i=1}^{n-1}|X_i - X_{i+1}|$? Yes:

$$\mathbb{E}[\sum_i |X'_i - X'_{i+1}| - \sum_i |X_i - X_{i+1}|] = \frac{1}{3}\#\{i : X'_i = X'_{i+1}\}.$$

If we assume that $\mathbb{P}[X_i = X_{i+1}] = \mathbb{P}[X_i = X_j | i \neq j]$, the following holds:

$$\mathbb{E}[\#\{i : X'_i = X'_{i+1}\}] = (n-1) \cdot \mathbb{P}[X_i = X_j | i \neq j] =$$

$$(n-1) \cdot \sum_z \mathbb{P}[X_i = X_j = z | i \neq j] =$$

$$(n-1) \cdot \sum_z \mathbb{P}[X_i = z]\mathbb{P}[X_j = z | i \neq j, X_i = z] =$$

$$(n-1) \sum_z \frac{Y_z}{n} \frac{Y_z - 1}{n-1} = \frac{1}{n} \sum_z Y_z(Y_z - 1).$$

Then:

$$\mathbb{E}[\frac{1}{(n-1)} \sum_i |X'_i - X'_{i+1}| - \frac{1}{(n-1)} \sum_i |X_i - X_{i+1}|] =$$

$$\frac{1}{3(n-1)} \mathbb{E}[\#\{i : X'_i = X'_{i+1}\}] = \frac{\sum_z Y_z^2}{3n(n-1)} - \frac{1}{3(n-1)} =$$

$$\mathbb{E}[\frac{1}{n(n-1)} \sum_{i \neq j} |X'_i - X'_j| - \frac{1}{n(n-1)} \sum_{i \neq j} |X_i - X_j|].$$

Thus the expected deviation is the same in both cases. However the assumption that $\mathbb{P}[X_i = X_{i+1}] = \mathbb{P}[X_i = X_j | i \neq j]$ is not one we can easily make. Actually, when a differs from b, it is also likely that these probabilities are different. Including this however would probably only make the difference between a and b larger, so we may choose to ignore the described effect.

For the Zürich 4 data, two consecutive non-outlying observations are the same in 25 cases, so this gives us a share of $\frac{25}{2485} = \frac{5}{497} \approx 0,010$. Meanwhile $\sum_z Y_z^2 = 96627$, which gives us a share of $\frac{96627}{2486 \cdot 2485} - \frac{1}{2485} \approx 0,012$. That is close enough.