# Report on Exam Task
# Simulation and Modelling of Communication Networks

Nicolás Chopitea Kober, Sebastian Lindner

Summer Term 2016

# Contents

# 1 Overview

## 1.1 Description

We were tasked by a university to analyze the usefulness, practicability and limitations of a remote university building's direct radio link to the main university campus.

In this chapter we will attempt to fully capture the scenario at hand, abstract it into a model and extract the requirements that have to be fulfilled in order to have an applicable solution to connecting the remote building with the main campus via radio.

## 1.2 Requirement Analysis

### 1.2.1 Network Description

The aim is to connect a remote building's network to the main university campus network. The fast cabled connection is endangered due to a construction site in close proximity to the building. That is why a direct radio link could be employed to maintain connectivity throughout the construction process.

The following network usage cases could be identified:

**CCTV** A CCTV camera is connected to the remote building's router.
It is continuously transmitting a livestream of its video.

**Wireless** A wireless access point operating on the *802.11g* standard is connected to the remote building's router.
Connected to this access point are users engaged in the following three activities.

    **FTP Upload** A single file upload can be observed that is ongoing throughout the simulation of the network.

    **Video Livestream** A remotely located professor is livestreaming his or her lecture to a video conference laptop.
    This connection is bidirectional so that participating students can ask questions.

    **Web Browsing** A variable number of students browsing the web is connected to the access point at any time.

This describes the network inside the remote building that we are trying to connect. The main campus' network can currently be simplified to three connections:

**Remote Professor** A fast connection to the remotely located professor's laptop is established via the Deutsches Forschungsnetz.

It will communicate with the video conference laptop inside the remote building.

**Porter's Office** A CCTV monitoring station sits in the porter's office.

The livestream of the remote building's CCTV camera will travel here.

**Internet** A VDSL connection to the Internet is present.

Both web browsing and FTP upload traffic will have this destination.

A graphical representation of the scenario is as follows and includes technical details such as channel delay and bandwidth:
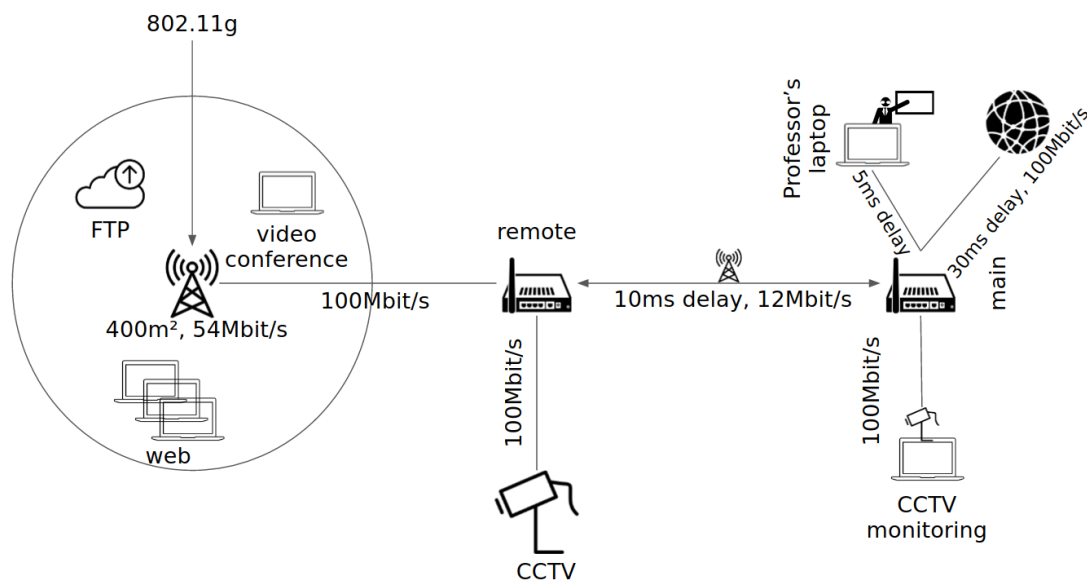


Figure 1.1: A model of the scenario depicted

## 1.2.2 Statistical Web Browsing Model

The student's web browsing behavior is difficult to capture. We are going to model it by assuming a student issues an HTTP request, receives a response and then spends some time reading the response that is exponentially distributed. Missing at this point is the size of the response that follows a request. To model this we have analyzed a trace file containing 1000 response size values.

We have chosen the *chi squared goodness of fit test* to evaluate how well a distribution fits the observed data. As a first step we investigated the density of values within request size intervals. These intervals initially were of equal size and the observed values were associated to an interval that they lied in. To remove intervals with no values associated to them, these were merged with neighboring intervals. In this way we obtained 160

intervals of which most have observed values associated to them. This is a graphical representation of what we found:
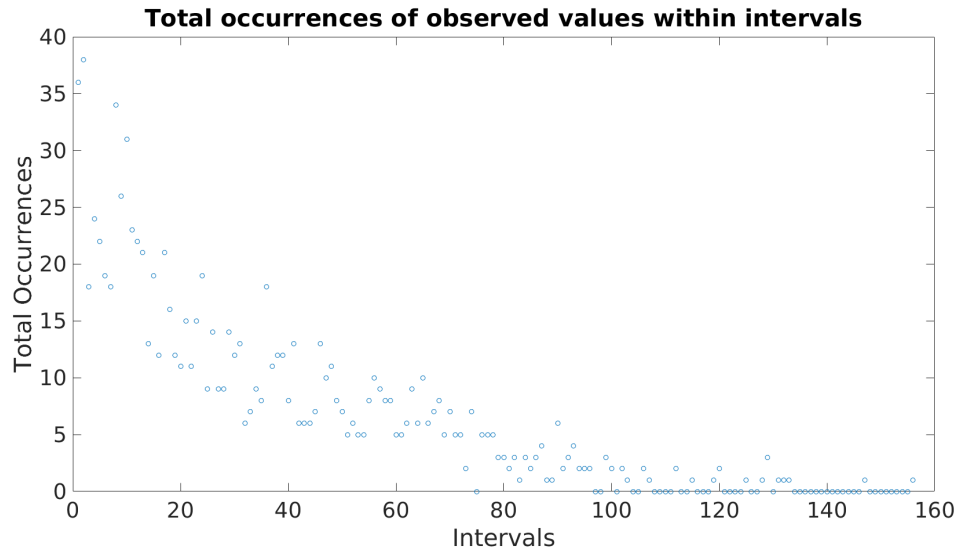


Figure 1.2: Request size value density in intervals

We have the highest density of observed values in regions of low response sizes, and the number of observations decreases exponentially for increasing response sizes. This looks closely related to values coming from a *negative exponential distribution*, which is why we decided to apply the test against this distribution.

The test states that the observed values follow an Exponential distribution with mean $\lambda = 580390$ Byte at a significance level of 99.95%.

### 1.2.3 Open Questions

We have now modeled the scenario at hand. What remains unclear is

1. What number of students can the network handle?

2. What are the bottlenecks of the network?

3. How is the lecture livestream's error rate correlated to the number of students?

4. What impact does the CCTV livestream have on the network?

5. What impact does the FTP upload have on the network?

In the next chapter we will try and answer these questions.

# 2 Simulation

We are going to answer the open questions from section 1.2.3 through simulation. For this we have modeled the scenario using the *omnet++* simulator.

## 2.1 Configuration

We have set up the simulations in the following ways.

**Simulation Time** Deciding upon a reasonable simulation time is essential. There are several different protocols in use here. The *802.11g WLAN standard* has, for example, a certain time that it takes until handshakes are complete and regular transmission can start.

That is why we have simulated the scenario for an extensive period of 5000s to see how throughput fluctuates in which time frames. Figure 2.1 shows the throughput of the remotely located Professor's laptop that livestreams his lecture to the remote building's conference laptop. We chose this use case as it represents the most time critical of all:
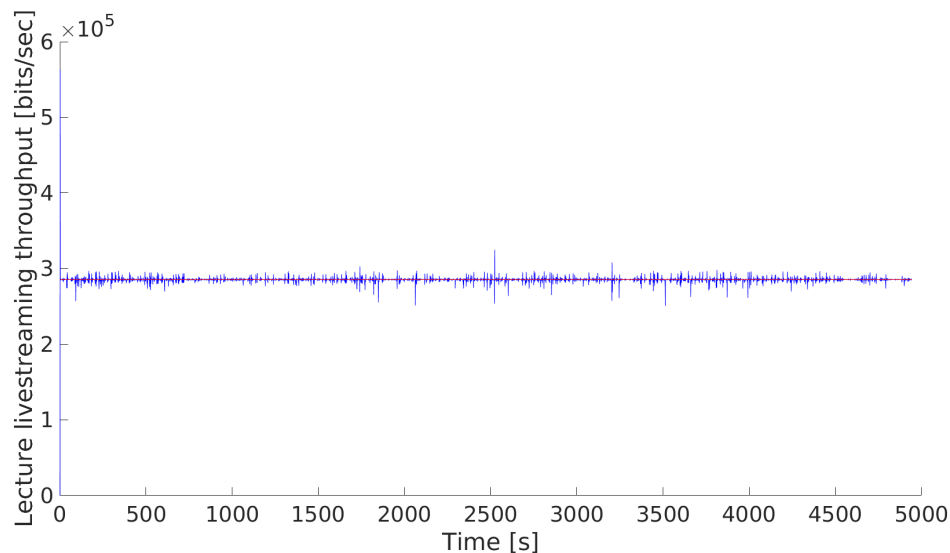


Figure 2.1: Throughput over 5000s

Our approach to this problem looks like this:

We want to find a simulation time $t$ so that if we went through the throughput depicted in figure 2.1 and sliced it into time intervals of length $t$ - in a sliding window-like manner, and then averaged the throughput values inside the sliding windows, we want these means to be relatively close to each other. This would mean that our $t$ is large enough so that it does not matter much which sliding window we took, the result would be more or less the same. This is equivalent to looking at the variance of these mean values. We applied this procedure on the data from the 5000s run for all $t \in \{250s, 350s, 500s, 750s, 1000s\}$. The result can be seen in figure 2.2.
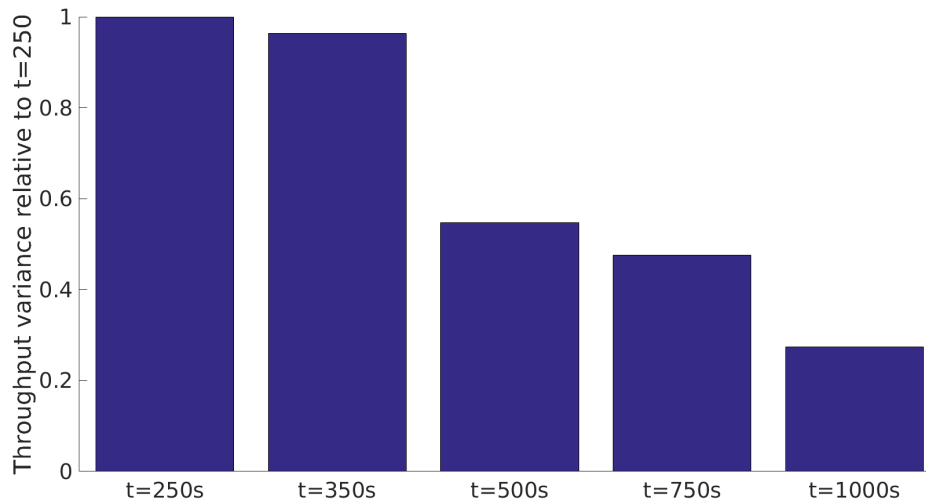
Figure 2.2: Relative variance of sliding window throughput means, where windows were of length $t$, on 5000s data

One can observe that doubling $t$ from $t = 250s$ to $t = 500s$ roughly halves the variance. Increasing it further greatly increases the time required to run these simulations while gaining relatively little.

This led us to the decision of setting the simulation time $t = 500s$.

We investigated the *warm-up period* of the WLAN protocol over long runs with $t \geq 500s$. We could not detect any significant warm-up period on time scales of this magnitude and conclude that its impact is negligible.

**Number of Repetitions** As simulations rely on random events and those build upon pseudo-random numbers that we generate, it is important to run each and every simulation several times with new seeds for the random number generators in order to cover a meaningful number of possibilities with the goal of approximating real world behavior.

Each run's performance is recorded and denoted as a *sample.* Following the *Central Limit Theorem* we can assume a sample mean to stem from a *Normal distribution.*

A sample mean estimate

$$\mu_m = \frac{1}{m} \sum_{j=1}^{m} y_j$$

where $m$ = number of samples
and $y_i = \text{i}^{\text{th}}$ sample
has a confidence interval of

$$\mu_m - \frac{z \cdot \sigma}{\sqrt{m}} \leq \mu \leq \mu_m + \frac{z \cdot \sigma}{\sqrt{m}}$$

where $\sigma$ = Normal distribution's variance
and $z$ is found using a Normal distribution's table using the required level of confidence $1 - \alpha$, $\alpha \in [0, 1]$.

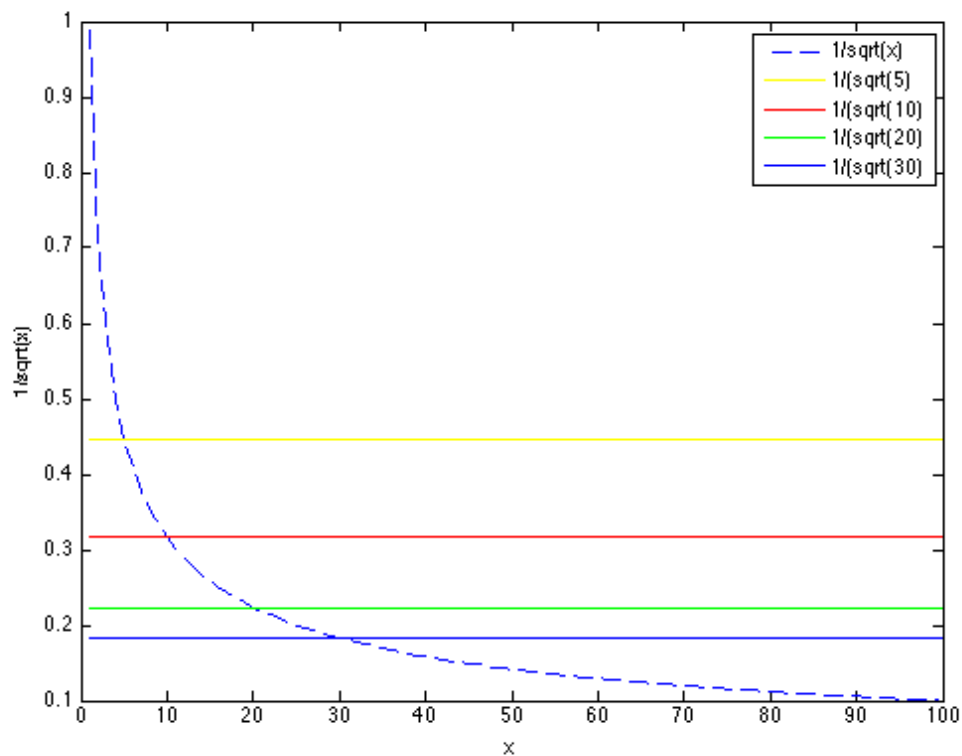So apparently the confidence interval depends on $\frac{1}{\sqrt{m}}$, which looks like this:



Figure 2.3: Behavior of $\frac{1}{\sqrt{x}}$

In general it should hold that the more repetitions your simulation goes through, the closer its behavior gets to real world conditions. But as with many things, we want to find a good balance between *correctness* and the *required effort*, or a good efficiency in other words. From investigating figure 2.3 we concluded that setting $m = 20$ seems like a good trade-off as at that point increasing $m > 20$ further does not yield much better results anymore, relative to values of $m < 20$ where performance gains are more significant.

**Number of students** As the number of students is variable we simulated for a number of values, narrowing down the maximum number of students that could be supported when taking all requirements into account.

**CCTV Camera** It was communicated to us that turning off the CCTV camera at the remote building location is an option. Thus we simulated every scenario once with the camera *on* and once with it being *off*.

## 2.2 Observation

This section covers the observations we can make when simulating the network in different scenarios. Error bars always represent a confidence level of $1 - \alpha = 0.95$.

### 2.2.1 Lecture Livestreaming Error Rate

The most time critical application is that of the bidirectional lecture video stream. We will investigate it in this section.

The livestream demands that the packet delay remains $< 100$ms. Packets that arrive later than that are dropped and contribute to the error rate. This error rate must stay below 5% in order to have a functioning livestream.

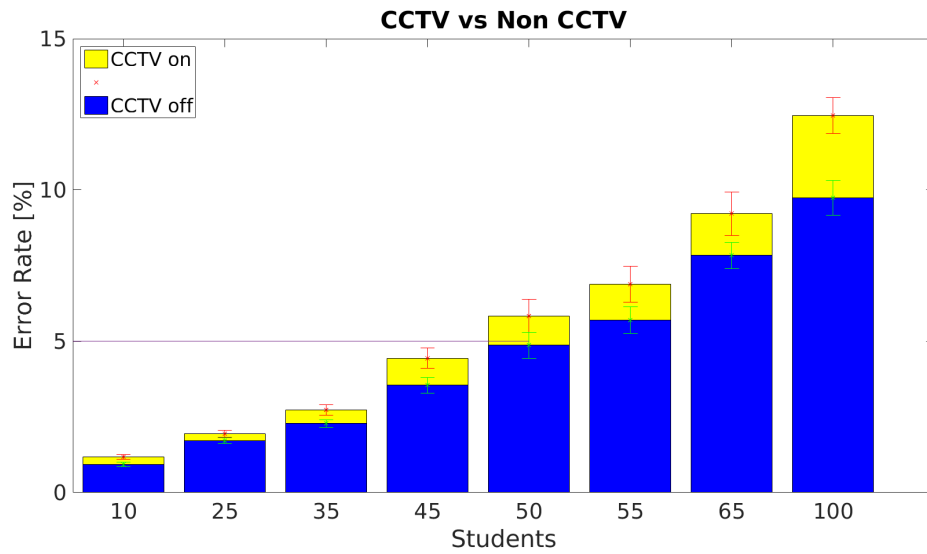Figure 2.4 shows the behavior of the error rate.

Figure 2.4: Lecture Livestream Error Rate

As we can see, the cut-off point seems to lie at $n = 50$ students. For an online CCTV camera the error rate lies just barely outside the allowed range. For an offline camera the livestream can still go with the occasional hiccup: the confidence interval tells us that some runs' mean error rates lied beyond the 5% mark and some did not.

Building upon this result we decided to focus our simulations on $n \in \{35, 45, 55, 65\}$ number of students as this set covers the critical part of the scenario, namely that range where the livestream error rate starts going beyond the 5% mark.

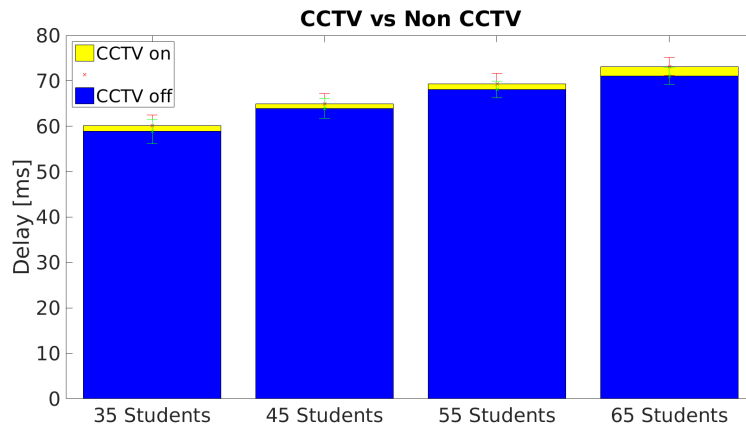Figure 2.5 shows the average delay of the connection.



Figure 2.5: Lecture Livestream Average Delay

We observe no surprises here. The delay increases for more users sharing the network and a turned on camera marginally increases the delay.

## 2.2.2 Web Browsing Behavior

The scenario's variability stems from an unknown number of students that simultaneously browses the web. Building upon results from section 2.4 we decided to focus on $n \in \{35, 45, 55, 65\}$ students so that the range around $n = 50$ where the lecture livestream starts having performance problems is depicted.
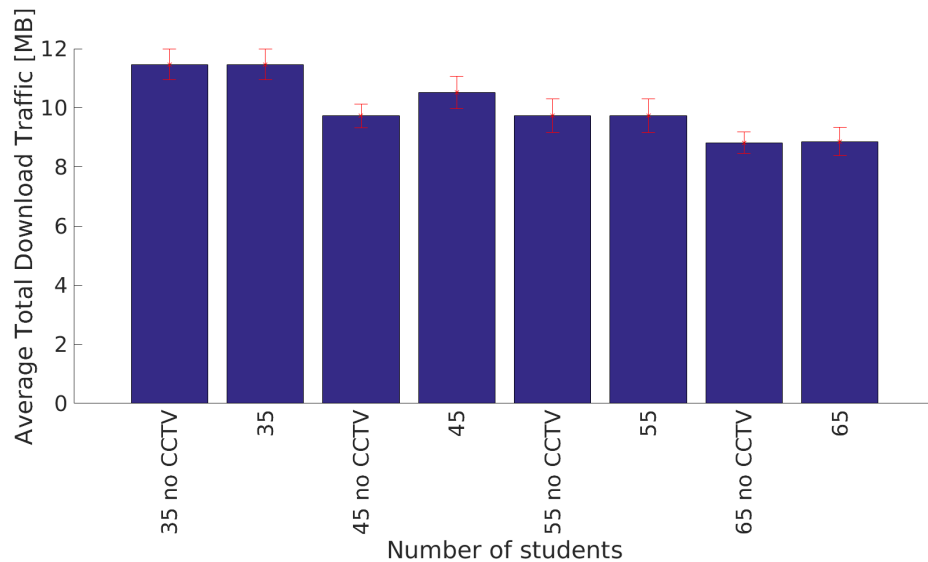


Figure 2.6: Total download traffic of an average student over 500s

Figure 2.6 shows us the averaged total download traffic of any student in the eight different simulations. As expected the total decreases for more users.

What is interesting to observe is that turning the CCTV camera on or off does not influence the total in a significant way. For $n = 45$ the total actually increases for an online camera, which seems counterintuitive at first glance. We will investigate this further in chapter 3.

## 2.2.3 File Uploading Behavior

The volume of data that the FTP client uploads during the simulation looks like this:

Figure 2.7: Total upload traffic of the FTP client over 500s
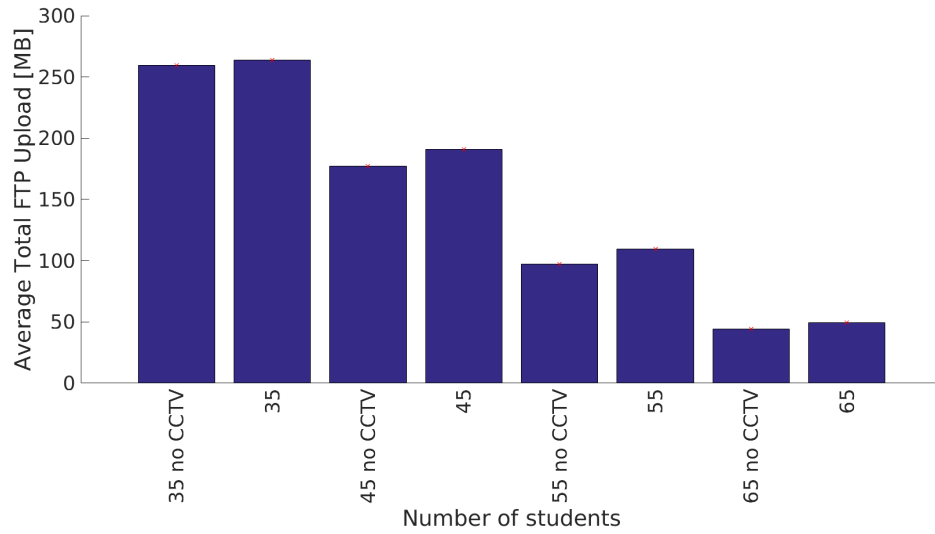
As one could expect, if more users share the network then the FTP client will have less chances of sending out its data. Interestingly the client is able to push out marginally more data if the CCTV camera is online. A more in depth analysis of this can be found in section 3.2.1.
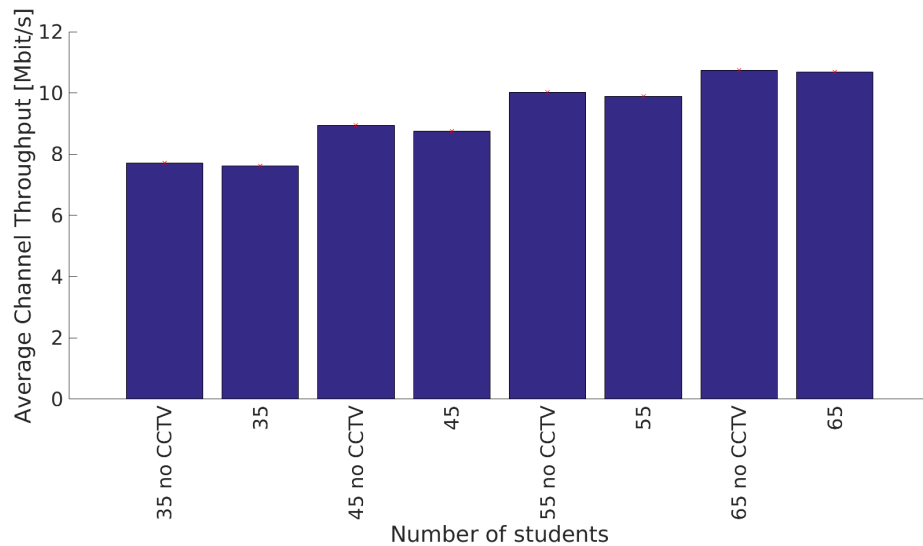
### 2.2.4 Backup Channel Utilization



Figure 2.8: Average Channel Throughput over 500s

As we can see, the average channel utilization is below its maximum of $12\frac{\text{Mbit}}{s}$ even for $n = 65$ students which is a use case that can not support a lecture livestream error rate below 5%.

# 3 Analysis

We have made many interesting observations in section 2.2. In this chapter we will try to analyze these observations, interpret them, evaluate the usefulness of the backup radio channel and give ideas on how to improve the scenario.

## 3.1 Supported Number of Students

As section 2.2.1 showed us, more than $n = 50$ students can certainly not be supported, even if the camera is turned off, if we want to maintain a lecture livestream's error rate of $< 5\%$. If the stream becomes noticeably erroneous, then we advise the administrator to turn off the CCTV camera. If the number of students goes far beyond $n \geq 45$, turning off the camera results in the system being able to support an additional, small number of students. Staying below this threshold, running the CCTV camera should not be a problem. Approaching $n = 55$ students, given a turned off CCTV camera, the livestream will start going beyond the critical $5\%$ error rate and start having a notable packet loss.

## 3.2 Radio Link Traffic

Many inputs to the backup radio link are known to us. Only the number of students is variable here. This section aims at breaking down what contributes to the backup radio link's traffic.

### 3.2.1 CCTV Camera's Influence on the Generated Traffic

As observed in section 2.6 the CCTV camera does not appear to affect the wireless users' throughput in a significant way. Lets recall the relevant network setup:
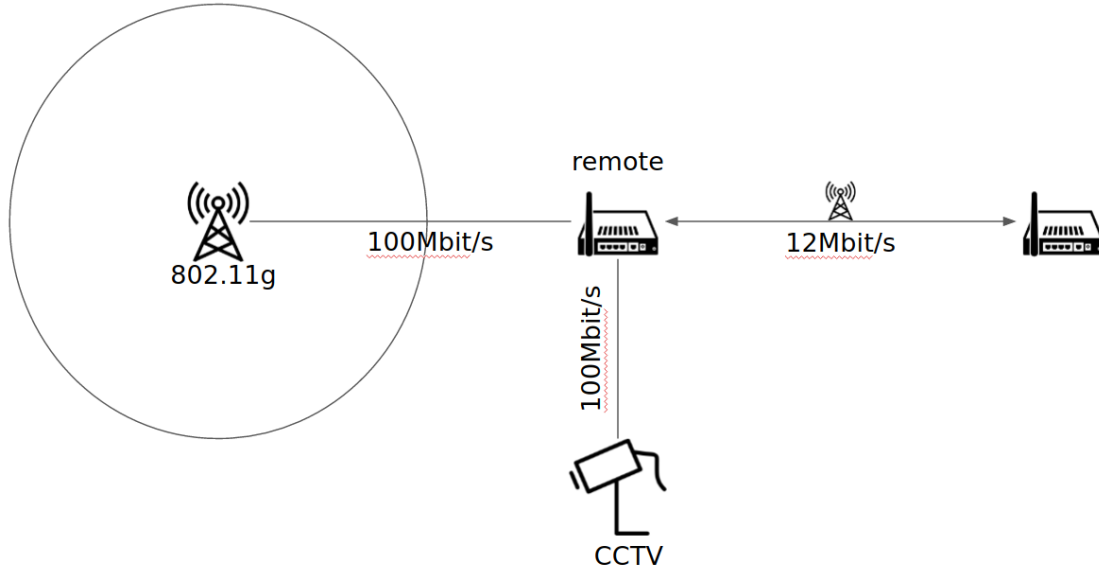
Figure 3.1: Network model excerpt - CCTV and WLAN

Apparently CCTV and the WLAN subnetworks do not directly share a link. Their data meets at the backup radio link as both data streams travel out towards the main campus. We know the CCTV's data rate of

$$d_{\text{CCTV}} = \frac{10\text{kB}}{40\text{ms}} = \frac{2\text{MBit}}{\text{s}}$$

We also know that the backup channel supports a bandwidth of

$$B_{\text{CHN}} = \frac{12\text{Mbit}}{\text{s}}$$

From this follows that the CCTV data stream takes up $\frac{1}{6}$ of the backup channel's bandwidth, leaving a majority of the bandwidth

$$B_{\text{CHN}}^{\text{free}} = \frac{5}{6} B_{\text{CHN}} = \frac{1250\text{kB}}{\text{s}}$$

free for the WLAN subsystem and so we can set the influence of the camera on the traffic as:

$$d_{\text{CCTV}} = \frac{1}{6} \cdot B_{\text{CHN}}$$

Interestingly, turning on the CCTV camera sometimes results in HTTP and FTP traffic totals increasing. Figure 2.8 shows us that in general the backup channel bandwidth is sufficient to support all generated traffic. Consequently this constant increase in traffic caused by the CCTV camera is negligible when it comes to those clients' throughput. The difference between a use case with an online camera and one with an offline camera is very small and we can therefore say that the CCTV traffic has no significant influence on HTTP and FTP traffic.

### 3.2.2 Lecture Livestream's Influence on the Generated Traffic

We have knowledge about the technical details of the livestream:

$$d_{\text{STR}}^{\text{1-way}} = \frac{1400\text{B}}{40\text{ms}} = \frac{0.28\text{MBit}}{\text{s}}$$

$$\Rightarrow d_{\text{STR}} = 2 \cdot d_{\text{STR}}^{\text{1-way}} = \frac{0.56\text{MBit}}{\text{s}}$$

$$\Rightarrow d_{\text{STR}} = 0.0467 \cdot B_{\text{CHN}}$$

### 3.2.3 FTP and HTTP Influence on the Generated Traffic

The constant FTP upload that takes place is more difficult to capture. FTP is an application layer protocol that operates on TCP. As such it has no knowledge of the network's current state. It passes on as much data as it wants to send, leaving the job of breaking down this data into chunks that *can* actually be sent to the TCP protocol. Similarly the HTTP traffic has random elements. We have determined that

$$B_{\text{CHN}}^{\text{usage}} = d_{\text{STR}} + d_{\text{CCTV}} + d_{\text{HTTP}} + d_{\text{FTP}}$$

$$\Leftrightarrow$$
$$B_{\text{CHN}}^{\text{usage}} = 0.0467 \cdot B_{\text{CHN}} + \frac{1}{6} \cdot B_{\text{CHN}} + d_{\text{HTTP}} + d_{\text{FTP}}$$

which leaves for HTTP and FTP traffic

$$B_{\text{CHN}} - B_{\text{CHN}}^{\text{usage}} = 0.786633333 B_{\text{CHN}} - d_{\text{HTTP}} - d_{\text{FTP}}$$

and so

$$d_{\text{HTTP}} + d_{\text{FTP}} \leq \frac{4}{5} B_{\text{CHN}}$$

## 3.3 Conclusion

HTTP and FTP traffic makes up the bulk of the data transmitted over the backup radio link. CCTV and videostream data is no more than $\frac{1}{5}$ of the total traffic.

As figure 2.8 demonstrates the channel bandwidth is large enough to support the generated traffic on average. Even for a large number of students $n \geq 55$ the channel is - on average - not fully utilized.

The lecture livestream's error rate however still suffers from a number of students that large. The livestream has a notable error rate of larger than 5% on average for $n \geq 50$ if the CCTV camera is online, and for $n \geq 55$ if it is offline. From this we can conclude that turning off the camera will lead to an additional 5 students that can browse the web while fulfilling all requirements.

HTTP and FTP traffic behaves in an expected kind of way. HTTP download and FTP upload traffic decreases when more users share the network. FTP upload traffic suffers a bigger hit than the average HTTP client, which is explained by the WLAN protocol not prioritizing traffic and so the single FTP client's share of the time accredited to it becomes proportionally smaller compared to that of all the HTTP clients. Both HTTP and FTP performances remain in a reasonable window in all cases.

To sum it up, the backup channel seems sufficient. Its bandwidth is large enough to support the generated traffic. Its average utilization is below its maximum when the livestream error rate starts to shoot up, so the bottleneck is elsewhere.

## 3.4 Improvements

As we have found out the channel bandwidth is sufficient and the optional CCTV camera contributes a foreseeable, constant amount of data. To improve the results the time share each client in the WLAN subnetwork is given could be adjusted to account for *Quality of Service* for the livestream. There are several networking standards available that implement such a feature. *IEEE 802.1q* is just one of many and its technical description goes beyond the scope of this report. As we have learned in the previous sections, dropped packets that contribute to the error rate are not dropped because the channel is always fully utilized. Instead the share of the single livestream client in the subnetwork becomes too small for a growing number of total clients. The time needed for a packet to travel from one end of the livestream connection to the other thus increases because more WLAN clients want to be served. If its packets were prioritized and those of the HTTP and possibly the FTP client neglected in its favor, then we foresee a larger number of students that could simultaneously use the network, while meeting the required livestream error rate.