Universidad del País Vasco / Euskal Herriko Unibertsitatea

MASTER THESIS

Development and Testing of Index Policies in Internet Routers

Author:
Josu Doncel

Advisor:
Peter Jacko
Urtzi Ayesta

Leioa, September 15, 2011

Acknowledgements

I would like to express my gratitude to my supervisor, Dr. Peter Jacko, who has supported my throughout my master thesis with patience. I appreciate his vast knowledge and skill in many areas (e. g. Dynamic and Stochastic Resource Allocation, Bandit Problems, Markov Decision Processes and Optimization in Graphs and Networks) and his assistance he provided at all levels of the research project.

A very special thanks goes to my co-supervisor Dr. Urtzi Ayesta, who gave me the opportunity of start my research in networks area. He helped me in the first steps of this work, and without his motivation and encouragement I would not be possible getting the results obtained.

I am grateful to the people I had the opportunity to work with during my visit to Konstantin Avratchenkov in INRIA-Sophia Antipolis. Specially I would like to thank Alina Quereilhac, from Planete team (INRIA) for helping me implementating the model based on index policies in Network Simulator-3 simulator.

I would like to thank BCAM-Basque Centre for Applied Mathematics for letting me join in Networks researchline in an Intership Program and INRIA for funding my visit.

I wish to thank everybody with whom I have shared experiences in the master courses. We have experienced many things together this year there are going to be present in my mind for long.

I cannot finish without saying how grateful I am with my parents. They have always supported and encouraged me to do my best in all matters of life.

Abstract

In this thesis we address the problem of fast and fair transmission of flows in a router, which is a fundamental issue in networks like the Internet. We develop and test a new policy that can be implemented in Internet routers in order to efficiently manage the packets arriving to their buffers in order to prevent congestion of the network when users want to send too much data. We have formulated a single Transmission Control Protocol (TCP) flow as a Markov decision process and analyzed both theoretically and numerically the existence of optimal control policies of special structure. In particular, we can conclude that for a variety of parameters, TCP flows can be optimally controlled in routers by so-called index policies, but not always by threshold policies. We have also implemented index policies in Network Simulator-3 and tested in a simple topology their applicability in real networks.

Contents

List of Figures

Chapter 1

Introduction

The TCP protocol (Transmission Control Protocol) is one of the protocols of TCP/IP internet suite that is complemented by the Internet Protocol (IP). TCP is responsible of sending reliable and ordered data from a one computer to another computer.

In 1988, Van Jacobson Jacobson (1988) described an internetworking protocol for sharing packets that controls TCP using additive increasing and multiplicative decreasing of the sending rate. This paper has become the basis of all the algorithms for network congestion avoidance.

The information that a user wants to send is divided in packets. IP protocol is responsible of exchanging this packets using the information of the header. The header describes all the information that IP needs to be sent to the packet's destination correctly.

The number of packets injected in to the network is controlled by TCP. TCP is used in many internet applications such as the World Wide Web (WWW), E-mail, File Transfer Protocol, Secure Shell, peer-to-peer file sharing, and some streaming media applications.

The packets travel through a network composed of routers. Due to stochastic nature of user requests, packets tend to accumulate at routers. This creates long waiting times, but also losses of packets because of buffer overflows.

It is therefore important to study how routers should behave in order to maximize the number of successfully delivered packets and to eventually prevent the collapse of the Internet due to congestion.

1.1 Project Aim

We can summarize the main goal of this work as the following:

(i) Our objective is deciding the best action to avoid future congestion of the network, taking into account α-fairness criterion. To do that, we describe a new control technique for TCP protocol based on ”index” policies. The index measures the efficiency of transmitting or not transmitting the flow/packet in an internet router. The method we are presenting in this work tries to solve the problem of the fast and fair applications in TCP applications.

(ii) Understand how the decrease factor of TCP impacts the solution. Therefore, we are analyzing if the problem for different instances can be solved under index policies (i. e. the problem is indexable).

1.2 Main Contributions

The contributions of this work are the following:

- We model TCP protocol as a Markov Decision Process for additive increasing and multiplicative decreasing instances. The router will take decision of transmitting or not transmitting the flow, according to the information provided by the user.

- We investigate the differences in the solution changing all the parameters of the model. So, we are looking for the α-fairness according to the different parameters of each user Altman et al. (2008). This is done both analytically and numerically by the means of a newly desinged algortihm.

- We validate the theoretical findings through extensive simulations. Me have chosen ns-3 to study the efficiency of this modeling.

1.3 Structure of Thesis

The rest of the thesis is structured as follows:

In Chapter 2 we explain the TCP control problem we are considering. In this section, we also formulated the additive increasing and multiplicative decreasing TCP model as a Markov Decision Process.

In Chapter 3 we explain the main results of this thesis. First of all, we introduce the Indexing Algorithm that return us information about indexability of the problem we are studying. After that, we present some analytical results of different multiplicative instances of TCP. Finally, we show the results obtained numerically for more complicated MDPs.

We have implemented the model in ns-3 simulator and the obtained results are detailed in Chapter 4.

The conclusions and future work are summarized in Chapter 5. We also explain in this section several open problems.

Chapter 2

Problem Description

In this section we will describe the context of a method which tries to avoid packet/flow congestion on a router Jacko and Sanso (2010). In this method the router has to take a decision, such as transmitting or not transmitting the flow, taking into account only its own congestion information. We will formulate the problem as a Markov Decision Process and, therefore, we will optimize the congestion control of the flows in the network.

We will apply the method to Internet flows behaving according to the Transmission Control Protocol (TCP). TCP is the component of TCP/IP internet suite which ensures that the data a computer wants to send to another computer delivers correctly. Furthermore, the TCP protocol is the protocol that is used by most of Internet applications, such as the World Wide Web, email, remote administration and file transfer.

The flow generates certain reward for its receiver, if it is delivered, for which it needs to be transmitted by the router to achieve it. This reward will be defined by the generalized α-fairness in this thesis Avrachenkov et al. (2004). The main problem is that these flows are changing their input rate, so several flows together can exceed the buffer space of the router. Avrachenkov et al. (2001) That is why the router has to take the decision of transmitting or not transmitting the flow, in order to avoid future congestion.

2.1 Model Description

In Figure 2.1, we can observe the behaviour of our model. The user (sending host) sends in a given time slot a fraction of the flow that is represented by W^{send} packets and the router must take a decision of transmitting or not transmitting these packets. Therefore, the flow that the router sends depends on the decision $a(t)$ and is represented by $W^{a(t)} \leq W^{send}$.

The reward received from the server is also represented in the figure as the acknowledgement and, as it depends only on the flow received from the router, we will represent

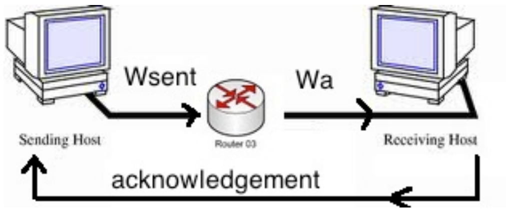


Figure 2.1: TCP sending packet example

it as $R^{a(t)}$.

The reward obtained in each time slot depends on the α we will use and the number of sending packets in that time slot is defined in the following way:

Definition 2.1. The reward vector from a received number of packets W_i depends on the α-fairness criterion and that is why we define it as follows:
$R = \frac{(1+W)^{1-\alpha}-1}{1-\alpha}$, if $\alpha \neq 1$, $\alpha \geq 0$
$R = \ln(1+W)$, if $\alpha = 1$
where W_i is the number of packets sent in one slot.

Now, we are going to explain how different increasing and decreasing TCP models are going to be represented in our model. In all instances, the $\boldsymbol{W}$ and $\boldsymbol{R}$ vectors will be constant and we will study the different instances of decreasing and increasing TCP model.

There are different increasing and decreasing models depending on the data sent in the previous time slot arrives correctly or not. In this work we will analyze different increase and decrease TCP models and their behavior by the Indexing Algorithm presented in Chapter 3.

Additive increase model sends one packet more if the data have arrived correctly to the destination. However, multiplying increase sends a number of packets multiplied by an increase factor in case that the data reaches with no error to the destination. In our TCP model, we consider the additive increase.

On the other hand, there are many decreasing models depending on the number of packets sent if the decided action is not transmitting the flow. In fact, we define restarting decrease as the model that consists on sending only one packet if action of not transmitting packet is applied. As we know from Jacko and Sanso (2010) the restarting decrease and additive increase model is indexable and can be optimally solved under threshold policy, so this case is considered just for completeness.

The most common decreasing models are based on a multiplicative decrease factor that we will call $\gamma \in [0, 1)$. If $\gamma = 0.5$, the decrease consists on sending half of the packets than in previous time slot and if $\gamma = 0$ the window is restarted to one.

We apply rounding to the decreasing factor to be used. We are always considering the floor of $currentstate$ and γ, except when the result is less than one. So if one packet was blocked, we again send one packet in order not to interrupt the connection.

The following formula shows how the decreasing behaves in our model:

$$\textit{nextstate} = \max\{floor(\gamma * \textit{currentstate}), 1\}$$

In our study, we will consider different decrease factor such as 0.5 or 0.3 and, also, we will observe what happens in case that the decrease is based on transmitting one packet less that the previous time slot (birth-death instance).

2.2 Formulation of Markov Decision Process Model

In this section, we start explaining some notation there is going to be used in TCP formulating as a MDP.

We present a binary-action Markov Decision Process (MDP) Puterman (1994) model for the router-based control of a single flow. The action space is $\mathcal{A}$={0,1}, depending if the action is transmitting or not transmitting packets and therefore the router must decide which action is the best with respect to given α-fairness criterion.

The state $X(t) = n$ represents that the number of packets sent in time slot t is n. That means that in state *i* the router transmits *i* packets if the action is 1 and if in state *i* the action is 0, it is not transmitting packets. Consequently, we can claim that the $\boldsymbol{W}^{sent} = \boldsymbol{W}^1 = (1, 2, ..., N)$, because action 1 consist on transmitting. However, since action 0 deals with not transmitting, we define $\boldsymbol{W}^0 = (0, 0, ..., 0)$.

We need to define the following parameters of the model before formulating it:

- The state space $\mathcal{N}$={1,2,..,N} as the set of states of the congestion window.
- The number of packets sent at state *n*, W_n^{sent}.
- The transition probability matrix if the router applies action *a* is $p_{n,m}^{a(t)|\pi}$:= the probability to move from state $n \in \mathcal{N}$ to $m \in \mathcal{N}$ if decided action is a(t) under policy π. We can observe $\boldsymbol{P}^1$ and $\boldsymbol{P}$ examples in the analytical and numerical computing section in Section 3.5.

- The bandwidth used is W_n^a and consists on the number of packets transmitted in state n if the router applies action a.

- The reward R_n^a is the expected reward obtained of W_n^a, according to the generalized α-fairness criterion.

Let ν be the Lagrangian multiplier that represents the per-packet transmission cost considered as a parameter of the router and let Π be the set of all history-dependent randomized policies for the flow. We must solve the following problem to find the best policy:

$$\max_{\pi \epsilon \Pi} \mathbb{E}_n^{\pi} \sum_{t=0}^{\infty} (\beta^t (R_{X(t)}^{a(t)} - \nu W_{X(t)}^{a(t)})), \tag{2.1}$$

where $\mathbb{E}_n^{\pi}$ is the expectation conditioned on the initial state $X(0) = n$ and on applying the policy π.

We define the discount factor β as the parameter that measures how much we care about future. It belongs to $(0, 1)$ and therefore, if t$\rightarrow \infty$, then $\beta^t \rightarrow 0$.

On the other hand, the parameter β we have just defined is commmonly used in other sense. We know that $1 - \beta$ can also be interpreted as the probability of finishing the flow. The appearance of β in our model causes that we can say we are creating a discounted model.

Using expectation properties in the above formula we are able to define the maximum policy depending only on the expected number of packets $\mathbb{W}_n^{\pi}$ and on the expected reward $\mathbb{R}_n^{\pi}$:

$$\begin{aligned} \max_{\pi \epsilon \Pi} \mathbb{E}_n^{\pi} (\sum_{t=0}^{\infty} \beta^t (R_{X(t)}^{a(t)} - \nu W_{X(t)}^{a(t)})) &= \max_{\pi \epsilon \Pi} \mathbb{E}_n^{\pi} (\sum_{t=0}^{\infty} \beta^t R_{X(t)}^{a(t)}) - \nu \mathbb{E}_n^{\pi} (\sum_{t=0}^{\infty} \beta^t W_{X(t)}^{a(t)})) = \\ &= \max_{\pi \epsilon \Pi} \mathbb{R}_n^{\pi} - \nu \mathbb{W}_n^{\pi} \end{aligned} \tag{2.2}$$

By Markov Decision Process theory, we know that there is an optimal policy such that π is stationary Puterman (1994). So, that let us consider that the policy π only depends on the current state.

Since the policy we are looking for only depends on the current state, the router can decide which action to apply depending only on the state. Therefore, beeing $\mathcal{S} \subseteq \mathcal{N}$, we define the policy $\mathcal{S}$ as the policy consisting on applying action zero (not transmitting) only in states in $\mathcal{S}$, and applying action one (transmitting) in states that do not belong to $\mathcal{S}$. That means that in $\mathcal{S}$ there are the states in which we transmit. That implies that we can write the previous formula in the following way:

$$\max_{\mathcal{S}\subseteq\mathcal{N}} \mathbb{R}_n^{\mathcal{S}} - \nu \mathbb{W}_n^{\mathcal{S}} \tag{2.3}$$

Chapter 3

Main Results

We are interested on solving problem (2.3), but we need to calculate $\mathbb{R}_n^\pi$ and $\mathbb{W}_n^\pi$ to compute the Index Algorithm that is going to be presented later.

3.1 How to Calculate $\mathbb{R}_n^\pi$ and $\mathbb{W}_n^\pi$?

In this section we are going to explain the manner of computing the $\mathbb{R}_n^\pi$ and the $\mathbb{W}_n^\pi$ values. We will describe it in the general case of any policy $\pi \in \Pi$ and any initial state $n \in \mathcal{N}$.

Let $a(t)$ be the action in time slot t and $p_{n,m}^{a(t)|\pi}$ the probability to move from state n to state m if it is decided action $a(t)$ under policy π.

We will start analyzing $\mathbb{R}_n^\pi$. We consider that in state $X(t)$ under policy π the reward generated is $R_{X(t)}^{a(t)}$.

$$\mathbb{R}_n^\pi = \mathbb{E}_n^\pi(\sum_{t=0}^{\infty} \beta^t R_{X(t)}^{a(t)}) = \mathbb{E}_n^\pi(R_n^{a(0)|\pi} + \beta \sum_{t=1}^{\infty} \beta^{t-1} R_{X(t)}^{a(t)}) =$$

$$= R_n^{a(0)|\pi} + \beta \sum_{X(1)\in\mathcal{N}} p_{n,X(1)}^{a(0)|\pi} \mathbb{E}_{X(1)}^\pi(\sum_{t=1}^{\infty} \beta^{t-1} R_{X(t)}^{a(t)}) = R_{X(0)}^{a(0)|\pi} + \beta \sum_{m\in\mathcal{N}} p_{n,m}^{a(0)|\pi} \mathbb{R}_m^\pi$$

In the last equality we are using the fact that in MDPs, only is taken into account the state. So, now it is very easy calculating $\mathbb{R}_n^\pi$ for each $m \in \mathcal{N}$ if we know the value of $\boldsymbol{P}^\pi$, a matrix consisting on $p_{n,m}^{a(t)|\pi}$ for all $m, n \in \mathcal{N}$, and $R^{a(t)}$ vector:

$$\mathbb{R}^\pi = B\ R^{a(t)|\pi},$$

where $B := (I - \beta \boldsymbol{P}^\pi)^{-1}$.

The value of $\mathbb{W}_n^\pi$ is computed in the same manner. Considering that in state $X(t)$

under policy $a(t)$ the number of packets sent is $W_{X(t)}^{a(t)}$, we have we following result:

$$\mathbb{W}_n^{\pi} = W_{X(0)}^{a(0)|\pi} + \beta \sum_{m \in \mathcal{N}} p_{n,m}^{a(0)|\pi} \mathbb{W}_m^{\pi}$$

So, what we have to compute to get the $\mathbb{W}_n^{\pi}$ is the following:

$$\mathbb{W}^{\pi} = B\, W^{a(t)|\pi},$$

where $B := (I - \beta \boldsymbol{P}^{\pi})^{-1}$ and $\boldsymbol{P}^{\pi}$ is a matrix consisting on $p_{n,m}^{a(t)|\pi}$ for all $m, n \in \mathcal{N}$.

3.2 Indexing Algorithm

Definition 3.1. We say that the problem (2.3) is indexable, if there exist real numbers ν_n, $n \in \mathcal{N}$ such that the following holds for every state $n \in \mathcal{N}$:

(i) if $\nu_n \geq \nu$, then it is optimal to transmit in state n and

(ii) if $\nu_n \leq \nu$, then it is optimal not to transmit in state n.

The function $n \to \nu_n$ is called the *index* and ν_n' s are called the *index* values.

The Indexing Algorithm returns the index that corresponds to each state if the problem is indexable. If the problem is not indexable, the problem (2.3) can not be solved by index policies. Therefore, there is nothing to conclude about this instance.

The description of Indexing Algorithm is as follows:

(i) Compute the start point ($\mathbb{R}_n^{\mathcal{S}}$,$\mathbb{W}_n^{\mathcal{S}}$) for $\mathcal{S} = \emptyset$.

(ii) Calculate the points ($\mathbb{R}_n^{\mathcal{S}\cup\{n\}}$, $\mathbb{W}_n^{\mathcal{S}\cup\{n\}}$) for all n$\notin$ $\mathcal{S}$ and the following two-point slope: ($R_n^{\mathcal{S}\cup\{n\}}$, $W_n^{\mathcal{S}\cup\{n\}}$) and ($\mathbb{R}_n^{\mathcal{S}}$,$\mathbb{W}_n^{\mathcal{S}}$).

(iii) Redefine $\mathcal{S} := \mathcal{S} \cup \{n^*\}$, where n^* is the state of $\mathcal{N}$ such that the slope is the highest, among $n \in \mathcal{S}$, and set

$$\nu_{n^*} := \frac{\mathbb{R}_{n^*}^{\mathcal{S}\cup\{n^*\}} - \mathbb{R}_{n^*}^{\mathcal{S}}}{\mathbb{W}_{n^*}^{\mathcal{S}\cup\{n^*\}} - \mathbb{W}_{n^*}^{\mathcal{S}}}$$

(iv) Go to step 2 until $\mathcal{S} = \mathcal{N} = \{1, 2, ..., N\}$.

We are interested in getting $\mathcal{S}(\nu)$ for every ν, such that solves problem (2.3). The $\mathcal{S}$ obtained in the algorithm is the set of states of the highest slope in order of appearance.

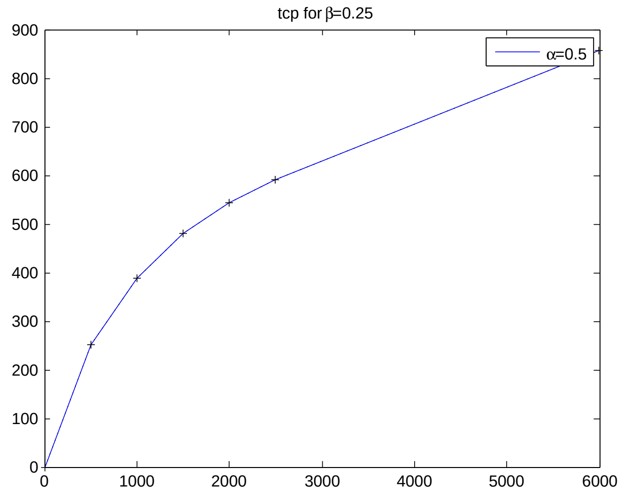


Figure 3.1: Points with highest slope for six states TCP model

Moreover, if the problem is indexable, computed $\mathcal{S}$ is the $\mathcal{S}(\nu)$ wanted and the slope computed connected to each state is the index that corresponds to that state.

In the algorithm described, we have obtained ν_n for all $n \in \mathcal{N}$ and the problem that we want to maximize has a fixed ν. According to Definition 3.1 we conclude that it is optimal not transmitting packets ($a(t) = 0$) if $\nu \geq \nu_n$ and it is optimal transmitting packets ($a(t) = 1$) if $\nu \leq \nu_n$.

In Figure 3.1 there is the output of the algorithm in a five states TCP model example that let us understand what the algorithm is realizing. We can observe that in each step it is only plotted the state with highest computed so that can be observed the solution of problem (2.3).

3.3 Checking Indexing Algorithm

According to Niño Mora (2002) there are two definitions that we can take into account to check whether our model is indexable or not: *PLC*- indexability and *LP*- indexability.

3.3.1 PCL-Checking

Definition 3.2. We say that problem (2.3) is *PCL*-indexable if it satisfies the following conditions:

(i) Positive marginal work: $w_i^{\mathcal{S}} > 0$ for $i \in \mathcal{N}$, where we know that $w_i^{\mathcal{S}} = \mathbb{W}_i^{<1,\mathcal{S}>} - \mathbb{W}_i^{<0,\mathcal{S}>}$

(ii) Monotone nonincreasing index computation: the index values produced by Indexing Algorithm satisfy

$$\nu^*_{i_1} \geq \nu^*_{i_2} \geq \dots \geq \nu^*_{i_n}$$

Let $x(t) = i$ be the current state of the model in time slot t. The first condition means that , as we said before, we consider a TCP model consisting in additive increase, so in our case we define the terms appeared there as follows:

$$\mathbb{W}_i^{<1,\mathcal{S}>} = W_i^1 + \beta \mathbb{W}^{\mathcal{S}}_{x(t+1)}$$

$$\mathbb{W}_i^{<0,\mathcal{S}>} = W_i^0 + \beta \mathbb{W}^{\mathcal{S}}_{x(t+1)}$$

We observe that both defined terms depend on $x(t+1)$ that consists on the state in the next time slot and we know that it will change if action 1 is decided or if action 0 is decided, so we can rewrite the definition above classifying them in the following cases:

- If $i \in \mathcal{S}$,

  $$\mathbb{W}_i^{<1,\mathcal{S}>} = W_i^1 + \mathbb{W}^{\mathcal{S}}_{i+1} = \mathbb{W}^{\mathcal{S}}_i$$

  $$\mathbb{W}_i^{<0,\mathcal{S}>} = W_i^0 + \mathbb{W}^{\mathcal{S}}_{x(t+1)_0} = \mathbb{W}^{\mathcal{S}}_{x(t+1)_0},$$

  where $x(t+1)_0$ is the state of the model that it goes decided action is 0 (not sending packets).

- If $i \notin \mathcal{S}$,

  $$\mathbb{W}_i^{<1,\mathcal{S}>} = W_i^1 + \mathbb{W}^{\mathcal{S}}_{i+1}$$

  $$\mathbb{W}_i^{<0,\mathcal{S}>} = W_i^0 + \mathbb{W}^{\mathcal{S}}_{x(t+1)_0} = \mathbb{W}^{\mathcal{S}}_i$$

  where $x(t+1)_0$ is also the state of the model that it goes decided action is 0 (not sending packets).

Notice that we consider $x(t+1)_1 = i+1$, due to considering additive increase.

Theorem 3.1. Niño Mora (2002) *If the conditions of* PCL*-indexability are satisfied in our model, the problem is indexable. However, if the conditions are not satisfied we can not say that is not indexable.*

We must continue with other checks in case the model is not under *PCL*-indexability conditions.

3.3.2 LP-Checking

We define *LP*-indexability as follows:

Definition 3.3. We say that problem (2.3) is *LP*-indexable if:

(i) $w_i^{\emptyset}$, $w_i^{\mathcal{N}} \geq 0$ for $i \in \mathcal{N}$

(ii) For each $\mathcal{S} \subseteq \mathcal{N}$, $w_i^{\mathcal{S}} > 0$ for $i \in \mathcal{N}$

(iii) For every wage $\nu \in \mathbb{R}$ there exists an optimal set $\mathcal{S} \subseteq \mathcal{N}$.

Lemma 3.1. *We only compute $w_i^{\mathcal{N}} \geq 0$ condition in the first condition of* LP*-indexability because $w_i^{\emptyset} \geq 0$ always holds.*

In fact, we use first condition of *LP*-indexability in the following way:

Theorem 3.2. *If the model does not satisfy the first condition of* LP*-indexability, the problem is not indexable.*

The last condition that we are going to check is that the sequence $\mathcal{S}$ appeared in the output of the algorithm is as follows: $\emptyset$, $\{1\}$, $\{1, 2\}$,...{1,2,...,N}. That means that the problem can be solved under a threshold policy and that is what we are interested in, as we explained before, due to we consider the additive increase on the TCP model.

3.3.3 Checking Indexing Algorithm

We describe the *Checking Indexing Algorithm*, that consist on the general Indexing Algorithm described before, but including all the checks we have just defined.

(i) Check that $w_i^{\mathcal{N}} \geq 0$ for $i \in \mathcal{N}$ holds (first condition of *LP*-indexability).

(ii) Compute the start point ($\mathbb{R}_n^{\mathcal{S}}$,$\mathbb{W}_n^{\mathcal{S}}$) for $\mathcal{S} = \emptyset$.

(iii) Calculate the points ($\mathbb{R}_n^{\mathcal{S}\cup\{n\}}$, $\mathbb{W}_n^{\mathcal{S}\cup\{n\}}$) for all n∉S and the following two-point slope: ($\mathbb{R}_n^{\mathcal{S}\cup\{n\}}$, $\mathbb{W}_n^{\mathcal{S}\cup\{n\}}$) and ($\mathbb{R}_n^{\mathcal{S}}$,$\mathbb{W}_n^{\mathcal{S}}$).

(iv) Redefine $\mathcal{S} := \mathcal{S} \cup \{n^*\}$, where n^* is the state of $\mathcal{N} - \mathcal{S}$ such that the slope is the highest.

(v) Check *PCL*-indexability for defined $\mathcal{S}$.

(vi) Go to step 2 until $S = \mathcal{N} = \{1, 2, ..., N\}$.

We conclude that there are different cases that can occur during the computation of the checked algorithm:

- *LP*-indexability is not true. That means the problem is not indexable.

- First condition of *LP*-indexability holds and *PCL*-indexability not. In this instance, we do not know if the problem is indexable or not.
- First condition of *LP*-indexability and *PCL*-indexability are true. The problem is indexable.
  - If the obtained $\mathcal{S}$ sequence grows one by one, the problem can be solved under threshold policy.
  - If the returned $\mathcal{S}$ of the algorithm does not grow one by one starting from $\emptyset$ ($\emptyset$, $\{1\}$, $\{1,2\}$,...{1,2,...,N}), the problem can not be solved under threshold policy.

3.4 Numerical Implementation

The algorithm presented for TCP protocol modeling needs a lot of computational resources such as matrix inverting and matrix and vector multiplying. Since the computer only uses a fixed number of digits, computing must have some numerical errors that we are going to discuss in this section.

The approximations of the calculation could suppose a problem we need to take into account in our work. Moreover, the problem becomes worst if depending on the tangent of the curve we are analyzing. If the tangent of the graph is high, small changes in x-axis causes big changes in y-axis. However, if the tangent is very small, small changes in y-axis causes big changes in x-axis.

As we can observe in Figure 3.1 the slopes are smaller when α is higher. So one of the problems that we have just explained could happen in our model. To avoid the effect of the approximation used in computer calculations, we define ϵ parameter as the value that let us conclude that numerical errors are not considered. This ϵ will be added to computed slopes in our algorithm.

In our instances, we have been trying many small ϵ parameters (from 10^{-6} to 10^{-12}) and the change in the return of the algorithm is the same.

To ensure that the numerical errors in the algorithm could be not considerable we implement the *reversed algorithm* that is based on the same idea that the Indexing Algorithm, but initial value of $\mathcal{S}$ set is $\mathcal{N}$ and instead of adding states to $\mathcal{S}$, it eliminates the best state found (we consider that the best state has the smallest slope). This new implementation of the algorithm also give us what we consider the same results comparing with the indexing algorithm, because the error is less than 10^{-6}.

Besides, we have been trying the Indexing Algorithm developed under $C++$ we are going to use in ns-3 simulation different float point data such as double and long double and in this case the errors obtained are less than 10^{-8}.

According to all the checks we have done connected to numerical instability we can conclude that significant numerical error should not be ocurring in our model.

3.5 Results

In this part of the work, we are presenting the results obtained from the Indexing Algorithm in the TCP model describe in Chapter 2.

3.5.1 Analytical computing of Index Values

Example 1: two states

In this example we detail all the computation related to two state instance for additive increase and any decrease model in order to understand better what the algorithm is doing.

We will assume that α and β are the parameters of our model. Therefore, the vectors and matrix described above will be in this case as follows:

- $\boldsymbol{W}^1 = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$
- $\boldsymbol{W}^0 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$
- $\boldsymbol{R}^1 = \begin{pmatrix} r_1 \\ r_2 \end{pmatrix}$
- $\boldsymbol{R}^0 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$
- $\boldsymbol{P}^1 = \begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix}$
- $\boldsymbol{P}^0 = \begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix}$

The matrices that we define in this instance are changing depending on the increase or decrease model we consider. As we are considering additive increase, the P^1 is fixed in this case.

Although P^0 matrix depends on the decreasing model we are analyzing, this matrix is unique for all decreasing model (any $\gamma \in [0,1)$) we assume because always goes to the first state if decided action is not transmitting packets, because the decreasing, at least, must be one and we have only two states.

The problem we are considering it is a additive increase and multiplicative decrease TCP model for all decrease factor γ. Hence, we know from Jacko and Sanso (2010) that the problem is indexable and it can be solved under threshold policy. We also can check that indices are equal to what we got in previous work.

What we are going to check in this case is that the algorithm give us the same results as Jacko and Sanso (2010).

In the first step, we calculate $\mathbb{W}$ and $\mathbb{R}$ for $\mathcal{S} = \emptyset$. We know that $W = W^0$, $R = R^0$ and $\boldsymbol{P} = \boldsymbol{P}^0$.

$$\mathbb{W}^{\mathcal{S}} = \left(I - \beta \boldsymbol{P}^{\mathcal{S}}\right)^{-1} W = \left[\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} - \beta \begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix}\right]^{-1} \begin{pmatrix} 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$\mathbb{R}^{\mathcal{S}} = (I - \beta \boldsymbol{P}^{\mathcal{S}})^{-1} R = \left[\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} - \beta \begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix}\right]^{-1} \begin{pmatrix} 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

In the second step, we compute again $\mathbb{W}^{\mathcal{S}}$ and $\mathbb{R}^{\mathcal{S}}$ for $\mathcal{S} = \{1\}$ and $\mathcal{S} = \{2\}$. In case of $\mathcal{S} = \{1\}$, we use the following:

$$W = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \ R = \begin{pmatrix} r_1 \\ 0 \end{pmatrix} \ \boldsymbol{P}^{\{1\}} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

In case of $\mathcal{S} = \{2\}$, we use the following vectors and matrix:

$$W = \begin{pmatrix} 0 \\ 2 \end{pmatrix}, \ R = \begin{pmatrix} 0 \\ r_2 \end{pmatrix} \ \boldsymbol{P}^{\{2\}} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

We calculate the slopes in both cases. The slopes are computed as follows:

$$\nu = \frac{\mathbb{R}_n^{\mathcal{S} \cup \{n\}} - \mathbb{R}_n^{\mathcal{S}}}{\mathbb{W}_n^{\mathcal{S} \cup \{n\}} - \mathbb{W}_n^{\mathcal{S}}}$$

But, as it can be observed, we need to know $\mathbb{R}^{\mathcal{S}}$ and $\mathbb{W}^{\mathcal{S}}$ for $S = \{1\}$ and $S = \{2\}$ to calculate it.

For $\{1\} \cup \{\emptyset\}$,

$$\mathbb{W}^{\{1\}} = \begin{pmatrix} \mathbb{W}_1^{\{1\}} \\ \mathbb{W}_2^{\{1\}} \end{pmatrix} = \left[\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} - \beta \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}\right]^{-1} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 & -\beta \\ -\beta & 1 \end{pmatrix}^{-1} \begin{pmatrix} 1 \\ 0 \end{pmatrix} =$$

$$= \begin{pmatrix} \frac{1}{1-\beta^2} & \frac{\beta}{1-\beta^2} \\ \frac{\beta}{1-\beta^2} & \frac{1}{1-\beta^2} \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} \frac{1}{1-\beta^2} \\ \frac{\beta}{1-\beta^2} \end{pmatrix}$$

$$\mathbb{R}^{\{1\}} = \begin{pmatrix} \mathbb{R}_1^{\{1\}} \\ \mathbb{R}_2^{\{1\}} \end{pmatrix} = \left[\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} - \beta \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \right]^{-1} \begin{pmatrix} r_1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 & -\beta \\ -\beta & 1 \end{pmatrix}^{-1} \begin{pmatrix} r_1 \\ 0 \end{pmatrix} =$$

$$= \begin{pmatrix} \frac{1}{1-\beta^2} & \frac{\beta}{1-\beta^2} \\ \frac{\beta}{1-\beta^2} & \frac{1}{1-\beta^2} \end{pmatrix} \begin{pmatrix} r_1 \\ 0 \end{pmatrix} = \begin{pmatrix} \frac{r_1}{1-\beta^2} \\ \frac{\beta r_1}{1-\beta^2} \end{pmatrix}$$

$$slope = \nu_1 = \frac{\mathbb{R}_1^{\emptyset \cup \{1\}} - \mathbb{R}_1^{\emptyset}}{\mathbb{W}_1^{\emptyset \cup \{1\}} - \mathbb{W}_1^{\emptyset}} = \frac{\frac{r_1}{1-\beta^2} - 0}{\frac{1}{1-\beta^2} - 0} = r_1$$

And for $\{2\} \cup \{\emptyset\}$,

$$\mathbb{W}^{\{2\}} = \begin{pmatrix} \mathbb{W}_1^{\{2\}} \\ \mathbb{W}_2^{\{2\}} \end{pmatrix} = \left[\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} - \beta \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \right]^{-1} \begin{pmatrix} 0 \\ 2 \end{pmatrix} = \begin{pmatrix} 1-\beta & 0 \\ 0 & 1-\beta \end{pmatrix}^{-1} \begin{pmatrix} 0 \\ 2 \end{pmatrix} =$$

$$\begin{pmatrix} \frac{1}{1-\beta} & 0 \\ 0 & \frac{1}{1-\beta} \end{pmatrix} \begin{pmatrix} 0 \\ 2 \end{pmatrix} = \begin{pmatrix} 0 \\ \frac{2}{1-\beta} \end{pmatrix}$$

$$\mathbb{R}^{\{2\}} = \begin{pmatrix} \mathbb{R}_1^{\{2\}} \\ \mathbb{R}_2^{\{2\}} \end{pmatrix} = \left[\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} - \beta \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \right]^{-1} \begin{pmatrix} 0 \\ r_2 \end{pmatrix} = \begin{pmatrix} 1-\beta & 0 \\ 0 & 1-\beta \end{pmatrix}^{-1} \begin{pmatrix} 0 \\ r_2 \end{pmatrix} =$$

$$\begin{pmatrix} \frac{1}{1-\beta} & 0 \\ 0 & \frac{1}{1-\beta} \end{pmatrix} \begin{pmatrix} 0 \\ r_2 \end{pmatrix} = \begin{pmatrix} 0 \\ \frac{r_2}{1-\beta} \end{pmatrix}$$

$$slope = \nu_2 = \frac{\mathbb{R}_2^{\emptyset \cup \{2\}} - \mathbb{R}_2^{\emptyset}}{\mathbb{W}_2^{\emptyset \cup \{2\}} - \mathbb{W}_2^{\emptyset}} = \frac{\frac{r_2}{1-\beta} - 0}{\frac{2}{1-\beta} - 0} = \frac{r_2}{2}$$

In this point, we notice that we do not know the values of the slopes, but taking into account definition Definition 2.1, we can say that $r_1 = \frac{2^{1-\alpha}-1}{1-\alpha}$ and $r_2 = \frac{3^{1-\alpha}-1}{1-\alpha}$. We will calculate the value of α for which $\nu_1 = \nu_2$ and in this manner we will conclude when $\nu_1 > \nu_2$ and $\nu_2 > \nu_1$.

$$\nu_1 = \nu_2 \Rightarrow r_1 = \frac{r_2}{2} \Rightarrow \frac{2^{1-\alpha} - 1}{1-\alpha} = \frac{1}{2} \frac{3^{1-\alpha} - 1}{1-\alpha}$$

We claim that the only root of the previous equation is $\alpha = 0$, according to the numerical check we have done. So, if $\alpha < 0$, then $\nu_1 < \nu_2$ and if $\alpha > 0$, then $r_1 > r_2$.

Lemma 3.2. *In two states TCP model with additive increase, always happens that $\nu_1 \geq \nu_2$.*

Now, we define $S := S \cup \{n^*\} = \emptyset \cup \{1\} = \{1\}$ and we continue with the algorithm analyzing the $\mathcal{N} = \{1, 2\}$ set:

$$\mathbb{W}^{\{1,2\}} = \begin{pmatrix} \mathbb{W}_1^{\{1,2\}} \\ \mathbb{W}_2^{\{1,2\}} \end{pmatrix} = \left[\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} - \beta \begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix} \right]^{-1} \begin{pmatrix} 1 \\ 2 \end{pmatrix} = \begin{pmatrix} 1 & -\beta \\ 0 & 1-\beta \end{pmatrix}^{-1} \begin{pmatrix} 1 \\ 2 \end{pmatrix} =$$

$$= \begin{pmatrix} 1 & \frac{\beta}{1-\beta} \\ 0 & \frac{1}{1-\beta} \end{pmatrix} \begin{pmatrix} 1 \\ 2 \end{pmatrix} = \begin{pmatrix} 1 + \frac{2\beta}{1-\beta} \\ \frac{2}{1-\beta} \end{pmatrix}$$

$$\mathbb{R}^{\{1,2\}} = \begin{pmatrix} \mathbb{R}_1^{\{1,2\}} \\ \mathbb{R}_2^{\{1,2\}} \end{pmatrix} = \left[\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} - \beta \begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix} \right]^{-1} \begin{pmatrix} r_1 \\ r_2 \end{pmatrix} = \begin{pmatrix} 1 & -\beta \\ 0 & 1-\beta \end{pmatrix}^{-1} \begin{pmatrix} r_1 \\ r_2 \end{pmatrix} =$$

$$= \begin{pmatrix} 1 & -\frac{\beta}{1-\beta} \\ 0 & \frac{1}{1-\beta} \end{pmatrix} \begin{pmatrix} r_1 \\ r_2 \end{pmatrix} = \begin{pmatrix} r_1 + \frac{r_2\beta}{1-\beta} \\ \frac{r_2}{1-\beta} \end{pmatrix}$$

$$slope = \nu_2 = \frac{\mathbb{R}_2^{\{1\}\cup\{2\}} - \mathbb{R}_2^{\{1\}}}{\mathbb{W}_2^{\{1\}\cup\{2\}} - \mathbb{W}_2^{\{1\}}} = \frac{\frac{r_2}{1-\beta} + \frac{\beta r_1}{1-\beta^2}}{\frac{2}{1-\beta} + \frac{\beta}{1-\beta^2}} = \frac{r_2 - \beta r_1 + \beta r_2}{2+\beta} = \frac{r_2 + \beta(r_2 - r_1)}{2 + \beta(2-1)}$$

In this example, we will assume the problem is *indexable* if there are non-increasing slopes. That is the same to say that the problem will be considered indexable if

$$\nu_1 \geq \nu_2.$$

We prove that the slopes are non-increasing in this two-state TCP additive increase model.

$$\nu_2 = \frac{r_2 - \beta r_1 + \beta r_2}{2+\beta} = \frac{r_2 + \beta r_1 - 2\beta r_1 + \beta r_2}{2+\beta} = \frac{r_2 + \beta r_1 - 2\beta(r_1 - \frac{r_2}{2})}{2+\beta} \leq$$

$$\leq \frac{r_2 + \beta r_1}{2+\beta} \leq \frac{2r_1 + \beta r_1}{2+\beta} = \frac{2+\beta}{2+\beta} r_1 = r_1 = \nu_1.$$

In the first inequality we use the fact that in the first step we proved that $r_1 \geq \frac{r_2}{2}$ and that implies that $-2\beta(r_1 - \frac{r_2}{2}) \leq 0$. And we use the same condition to claim that $r_2 \leq 2r_1$ in the second inequality.

Example 2: three states

In this example we detail all the computation conected to three state instance in order to prove that theorically it is possible to solve the Indexing Algorithm in case we are not in restarting case. Moreover, we will analyze numerically what is the behaviour of the model and we will observe that the behaviour is equal in both cases.

The previous example deals with explaining the main computation of the Indexing Algorithm and what should happen according to the theoretical point of view occurs. The goal of this instance, however, is proving that in theory the additive increase and non restarting TCP model of three states is always indexable and sometimes can be solved under threshold policy and we are going to prove that the numerical results satisfy the same conditions.

We also assume that α and β are parameters of our model. The vectors and matrix we using in this instance are described as follows:

- $\boldsymbol{W}^1 = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$
- $\boldsymbol{W}^0 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$
- $\boldsymbol{R}^1 = \begin{pmatrix} r_1 \\ r_2 \\ r_3 \end{pmatrix}$
- $\boldsymbol{R}^0 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$
- $\boldsymbol{P}^1 = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix}$
- $\boldsymbol{P}^0 = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$

Let's explain the likeness of this vectors and matrices. As we defined before, $\boldsymbol{W}^1$ and $\boldsymbol{W}^0$ do not change in the model, so in this case the size of the vector is three and it is similar to the previous example. Besides, $\boldsymbol{R}^1$ and $\boldsymbol{R}^0$ vectors also change according to the α that will be used, so r_1, r_2 and r_3 are parameters defined before in Definition 2.1. So, $\boldsymbol{R}^1$ is defined as before depending on r_1, r_2 and r_3 are parameters and $\boldsymbol{R}^0$ vector change depending on α too, but as $\boldsymbol{W}^0$ is zero, the reward vector is always zero.

The matrices that we define in this instance are 3×3 and they are also changing depending on the increase or decrease TCP model. We contemplate additive increase, therefore the $\boldsymbol{P}^1$ matrix is similar to what we described in example 1, but it is 3×3 as we defined before.

On the other hand, $\boldsymbol{P}^0$ matrix depends on the decreasing model we are considering. As we are in a three states TCP decreasing model and the decreasing must be at least one, we have two possible $\boldsymbol{P}^0$ matrices:

$$\boldsymbol{P}^0 = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}, \quad \boldsymbol{P}^0 = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$

The first one of them is the restarting case and we are not analizying because we know it is indexable from Jacko and Sanso (2010) and the index values are the following:

- $\nu_1 = r_1$
- $\nu_2 = \frac{r_2+\beta(r_2-r_1)}{2+\beta(2-1)}$
- $\nu_3 = \frac{r_3+\beta(r_3-r_2)+\beta^2(r_3-r_1)}{3+\beta(3-2)+\beta^2(3-1)}$

We must define the value of the decrease factor γ for which non-restarting decrease occur. As we described in Section 2.1 we consider the floor of multiplying the position of the current state and the decrease factor. So, if the decrease factor is less than $\frac{2}{3}$, we will be in restarting instance. That is the reason that the decrease factor used in our three states TCP model is greater than this critical value of γ.

We start from $\mathcal{S} = \emptyset$ again and we compute in the same manner as in the first example $\mathbb{W}^{\{\emptyset\}}$ and $\mathbb{R}^{\{\emptyset\}}$.

$$\mathbb{W}^{\{\emptyset\}} = (I - \beta \boldsymbol{P})^{-1} W = \left[\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} - \beta \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \right]^{-1} \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

$$\mathbb{R}^{\{\emptyset\}} = (I - \beta \boldsymbol{P})^{-1} R = \left[\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} - \beta \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \right]^{-1} \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

In the second step, we compute again $\mathbb{W}$ and $\mathbb{R}$ for $\mathcal{S} = \{n\}$ for all $n \in \mathcal{N}$ and the slopes obtained are as follows:

$$\nu = \frac{\mathbb{R}_n^{\{n\}} - \mathbb{R}_n^{\emptyset}}{\mathbb{W}_n^{\{n\}} - \mathbb{W}_n^{\emptyset}}$$

First, we calculate $\mathbb{R}_n^{\{n\}}$ and $\mathbb{W}_n^{\{n\}}$ for all $n \in \mathcal{N}$.

For $\mathcal{S} = \{1\}$,

$$\mathbb{W}^{\{1\}} = \begin{pmatrix} \mathbb{W}_1^{\{1\}} \\ \mathbb{W}_2^{\{1\}} \\ \mathbb{W}_3^{\{1\}} \end{pmatrix} = \left[\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} - \beta \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \right]^{-1} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 & -\beta & 0 \\ -\beta & 1 & 0 \\ 0 & -\beta & 1 \end{pmatrix}^{-1} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} =$$

$$= \begin{pmatrix} \frac{1}{1-\beta^2} & \frac{\beta}{1-\beta^2} & 0 \\ \frac{\beta}{1-\beta^2} & \frac{1}{1-\beta^2} & 0 \\ \frac{\beta^2}{1-\beta^2} & \frac{\beta}{1-\beta^2} & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} \frac{1}{1-\beta^2} \\ \frac{\beta}{1-\beta^2} \\ \frac{\beta^2}{1-\beta^2} \end{pmatrix}$$

$$\mathbb{R}^{\{1\}} = \begin{pmatrix} \mathbb{R}_1^{\{1\}} \\ \mathbb{R}_2^{\{1\}} \\ \mathbb{R}_3^{\{1\}} \end{pmatrix} = \left[\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} - \beta \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \right]^{-1} \begin{pmatrix} r_1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 & -\beta & 0 \\ -\beta & 1 & 0 \\ 0 & -\beta & 1 \end{pmatrix}^{-1} \begin{pmatrix} r_1 \\ 0 \\ 0 \end{pmatrix} =$$

$$= \begin{pmatrix} \frac{1}{1-\beta^2} & \frac{\beta}{1-\beta^2} & 0 \\ \frac{\beta}{1-\beta^2} & \frac{1}{1-\beta^2} & 0 \\ \frac{\beta^2}{1-\beta^2} & \frac{\beta}{1-\beta^2} & 1 \end{pmatrix} \begin{pmatrix} r_1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} \frac{r_1}{1-\beta^2} \\ \frac{\beta r_1}{1-\beta^2} \\ \frac{\beta^2 r_1}{1-\beta^2} \end{pmatrix}$$

$$slope = \nu_1 = \frac{\mathbb{R}_1^{\emptyset\cup\{1\}} - \mathbb{R}_1^{\emptyset}}{\mathbb{W}_1^{\emptyset\cup\{1\}} - \mathbb{W}_1^{\emptyset}} = \frac{\frac{r_1}{1-\beta^2} - 0}{\frac{1}{1-\beta^2} - 0} = r_1$$

And for $\mathcal{S} = \{2\}$,

$$\mathbb{W}^{\{2\}} = \begin{pmatrix} \mathbb{W}_1^{\{2\}} \\ \mathbb{W}_2^{\{2\}} \\ \mathbb{W}_3^{\{2\}} \end{pmatrix} = \left[\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} - \beta \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \right]^{-1} \begin{pmatrix} 0 \\ 2 \\ 0 \end{pmatrix} = \begin{pmatrix} 1-\beta & 0 & 0 \\ 0 & 1 & -\beta \\ 0 & -\beta & 1 \end{pmatrix}^{-1} \begin{pmatrix} 0 \\ 2 \\ 0 \end{pmatrix} =$$

$$= \begin{pmatrix} \frac{1-\beta^2}{1-\beta-\beta^2+\beta^3} & 0 & 0 \\ 0 & \frac{1-\beta}{1-\beta-\beta^2+\beta^3} & \frac{\beta(1-\beta)}{1-\beta-\beta^2+\beta^3} \\ 0 & \frac{\beta(1-\beta)}{1-\beta-\beta^2+\beta^3} & \frac{1-\beta}{1-\beta-\beta^2+\beta^3} \end{pmatrix} \begin{pmatrix} 0 \\ 2 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ \frac{2(1-\beta)}{1-\beta-\beta^2+\beta^3} \\ \frac{2\beta(1-\beta)}{1-\beta-\beta^2+\beta^3} \end{pmatrix}$$

$$\mathbb{R}^{\{2\}} = \begin{pmatrix} \mathbb{R}_1^{\{2\}} \\ \mathbb{R}_2^{\{2\}} \\ \mathbb{R}_3^{\{2\}} \end{pmatrix} = \left[\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} - \beta \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \right]^{-1} \begin{pmatrix} 0 \\ r_2 \\ 0 \end{pmatrix} = \begin{pmatrix} 1-\beta & 0 & 0 \\ 0 & 1 & -\beta \\ 0 & -\beta & 1 \end{pmatrix}^{-1} \begin{pmatrix} 0 \\ r_2 \\ 0 \end{pmatrix} =$$

$$= \begin{pmatrix} \frac{1-\beta^2}{1-\beta-\beta^2+\beta^3} & 0 & 0 \\ 0 & \frac{1-\beta}{1-\beta-\beta^2+\beta^3} & \frac{\beta(1-\beta)}{1-\beta-\beta^2+\beta^3} \\ 0 & \frac{\beta(1-\beta)}{1-\beta-\beta^2+\beta^3} & \frac{1-\beta}{1-\beta-\beta^2+\beta^3} \end{pmatrix} \begin{pmatrix} 0 \\ r_2 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ \frac{r_2(1-\beta)}{1-\beta-\beta^2+\beta^3} \\ \frac{r_2\beta(1-\beta)}{1-\beta-\beta^2+\beta^3} \end{pmatrix}$$

$$slope = \nu_2 = \frac{\mathbb{R}_2^{\{2\}} - \mathbb{R}_2^{\emptyset}}{\mathbb{W}_2^{\{2\}} - \mathbb{W}_2^{\emptyset}} = \frac{\frac{r_2(1-\beta)}{1-\beta-\beta^2+\beta^3} - 0}{\frac{2(1-\beta)}{1-\beta-\beta^2+\beta^3} - 0} = \frac{r_2}{2}$$

And for $\mathcal{S} = \{3\}$,

$$\mathbb{W}^{\{3\}} = \begin{pmatrix} \mathbb{W}_1^{\{3\}} \\ \mathbb{W}_2^{\{3\}} \\ \mathbb{W}_3^{\{3\}} \end{pmatrix} = \left[\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} - \beta \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \right]^{-1} \begin{pmatrix} 0 \\ 0 \\ 3 \end{pmatrix} = \begin{pmatrix} 1-\beta & 0 & 0 \\ -\beta & 1 & 0 \\ 0 & 0 & 1-\beta \end{pmatrix}^{-1} \begin{pmatrix} 0 \\ 0 \\ 3 \end{pmatrix} =$$

$$= \begin{pmatrix} \frac{1}{1-\beta} & 0 & 0 \\ \frac{\beta}{1-\beta} & 1 & 0 \\ 0 & 0 & \frac{1}{1-\beta} \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \frac{3}{1-\beta} \end{pmatrix}$$

$$\mathbb{R}^{\{3\}} = \begin{pmatrix} \mathbb{R}_1^{\{3\}} \\ \mathbb{R}_2^{\{3\}} \\ \mathbb{R}_3^{\{3\}} \end{pmatrix} = \left[\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} - \beta \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \right]^{-1} \begin{pmatrix} 0 \\ 0 \\ r_3 \end{pmatrix} = \begin{pmatrix} 1-\beta & 0 & 0 \\ -\beta & 1 & 0 \\ 0 & 0 & 1-\beta \end{pmatrix}^{-1} \begin{pmatrix} 0 \\ 0 \\ r_3 \end{pmatrix} =$$

$$= \begin{pmatrix} \frac{1}{1-\beta} & 0 & 0 \\ \frac{\beta}{1-\beta} & 1 & 0 \\ 0 & 0 & \frac{1}{1-\beta} \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ r_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \frac{r_3}{1-\beta} \end{pmatrix}$$

$$slope = \nu_3 = \frac{\mathbb{R}_3^{\{3\}} - \mathbb{R}_3^{\emptyset}}{\mathbb{W}_3^{\{3\}} - \mathbb{W}_3^{\emptyset}} = \frac{\frac{r_3}{1-\beta} - 0}{\frac{3}{1-\beta} - 0} = \frac{r_3}{3}$$

As the slopes depend on the values of r_1, r_2 and r_3, we do not know a priori which is the highest slope in this step. In Figure 3.2 we can solve the problem of finding the highest slope. In the picture, ν_1 is represented by the black line, ν_2 is the red line and ν_3 the blue line, where r_i, $i = 1, 2, 3$ are the values described in Definition 2.1 for $\alpha \neq 1$.

For $\alpha = 1$ instance, we know that $r_i = \log(w_i + 1) = \log i + 1$, for $i = 1, 2, 3$. So, it is

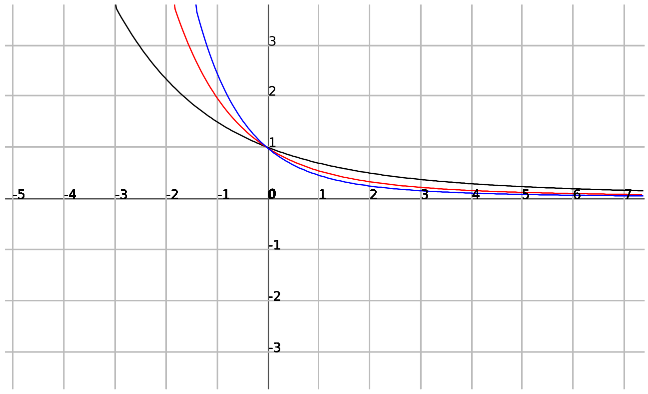


Figure 3.2: ν_1, ν_2 and ν_3 slopes in step one versus α

very easy to verify that happens the same as the $\alpha \neq 1$ case because $r_1 = \log 2 > \frac{r_2}{2} = \frac{\log 3}{2} > \frac{r_3}{3} = \frac{\log 4}{3}$. That is why we can claim the following lemma.

Lemma 3.3. *In three states TCP model with additive increase and non-restarting decrease, ν_1 is always greater than the other slopes in the first step.*

Now, we define $\mathcal{S} := \mathcal{S} \cup \{n^*\}$, where n^* is the state of the greater slope in the first step. So, we continue with the algorithm with $S = \{1\}$ calculating ν_2 and ν_3. As we said before, it is necessary to know the value of $\mathbb{R}_n^{\{1,n\}}$ and $\mathbb{W}_n^{\{1,n\}}$ for $n = 2, 3$ to do it.

For $n = 2$:

$$\mathbb{W}^{\{1\}\cup\{2\}} = \begin{pmatrix} \mathbb{W}_1^{\{1,2\}} \\ \mathbb{W}_2^{\{1,2\}} \\ \mathbb{W}_3^{\{1,2\}} \end{pmatrix} = \left[\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} - \beta \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}\right]^{-1} \begin{pmatrix} 1 \\ 2 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 & -\beta & 0 \\ 0 & 1 & -\beta \\ 0 & -\beta & 1 \end{pmatrix}^{-1} \begin{pmatrix} 1 \\ 2 \\ 0 \end{pmatrix} =$$

$$= \begin{pmatrix} 1 & \frac{\beta}{1-\beta^2} & \frac{\beta^2}{1-\beta^2} \\ 0 & \frac{1}{1-\beta^2} & \frac{\beta}{1-\beta^2} \\ 0 & \frac{\beta}{1-\beta^2} & \frac{1}{1-\beta^2} \end{pmatrix} \begin{pmatrix} 1 \\ 2 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 + \frac{2\beta}{1-\beta^2} \\ \frac{2}{1-\beta^2} \\ \frac{2\beta}{1-\beta^2} \end{pmatrix}$$

$$\mathbb{R}^{\{1\}\cup\{2\}} = \begin{pmatrix} \mathbb{R}_1^{\{1,2\}} \\ \mathbb{R}_2^{\{1,2\}} \\ \mathbb{R}_3^{\{1,2\}} \end{pmatrix} = \left[\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} - \beta \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}\right]^{-1} \begin{pmatrix} r_1 \\ r_2 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 & -\beta & 0 \\ 0 & 1 & -\beta \\ 0 & -\beta & 1 \end{pmatrix}^{-1} \begin{pmatrix} r_1 \\ r_2 \\ 0 \end{pmatrix} =$$

$$= \begin{pmatrix} 1 & \frac{\beta}{1-\beta^2} & \frac{\beta^2}{1-\beta^2} \\ 0 & \frac{1}{1-\beta^2} & \frac{\beta}{1-\beta^2} \\ 0 & \frac{\beta}{1-\beta^2} & \frac{1}{1-\beta^2} \end{pmatrix} \begin{pmatrix} r_1 \\ r_2 \\ 0 \end{pmatrix} = \begin{pmatrix} r_1 + \frac{r_2\beta}{1-\beta^2} \\ \frac{r_2}{1-\beta^2} \\ \frac{r_2\beta}{1-\beta^2} \end{pmatrix}$$

$$slope = \nu_2 = \frac{\mathbb{R}_2^{\{1\}\cup\{2\}} - \mathbb{R}_2^{\{1\}}}{\mathbb{W}_2^{\{1\}\cup\{2\}} - \mathbb{W}_2^{\{1\}}} = \frac{\frac{r_2}{1-\beta} + \frac{\beta r_1}{1-\beta^2}}{\frac{2}{1-\beta} + \frac{\beta}{1-\beta^2}} = \frac{r_2 - \beta r_1 + \beta r_2}{2+\beta}$$

For $n = 3$:

$$\mathbb{W}^{\{1\}\cup\{3\}} = \begin{pmatrix} \mathbb{W}_1^{\{1,3\}} \\ \mathbb{W}_2^{\{1,3\}} \\ \mathbb{W}_3^{\{1,3\}} \end{pmatrix} = \left[\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} - \beta \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix}\right]^{-1} \begin{pmatrix} 1 \\ 0 \\ 3 \end{pmatrix} = \begin{pmatrix} 1 & -\beta & 0 \\ 1-\beta & 1 & 0 \\ 0 & 0 & 1-\beta \end{pmatrix}^{-1} \begin{pmatrix} 1 \\ 0 \\ 3 \end{pmatrix} =$$

$$= \begin{pmatrix} \frac{1}{1-\beta^2} & \frac{\beta}{1-\beta^2} & 0 \\ \frac{\beta}{1-\beta^2} & \frac{1}{1-\beta^2} & 0 \\ 0 & 0 & \frac{1}{1-\beta} \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 3 \end{pmatrix} = \begin{pmatrix} \frac{1}{1-\beta^2} \\ \frac{1}{1-\beta^2} \\ \frac{3}{1-\beta^2} \end{pmatrix}$$

$$\mathbb{W}^{\{1\}\cup\{3\}} = \begin{pmatrix} \mathbb{R}_1^{\{1,3\}} \\ \mathbb{R}_2^{\{1,3\}} \\ \mathbb{R}_3^{\{1,3\}} \end{pmatrix} = \left[\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} - \beta \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix}\right]^{-1} \begin{pmatrix} r_1 \\ 0 \\ r_3 \end{pmatrix} = \begin{pmatrix} 1 & -\beta & 0 \\ 1-\beta & 1 & 0 \\ 0 & 0 & 1-\beta \end{pmatrix}^{-1} \begin{pmatrix} r_1 \\ 0 \\ r_3 \end{pmatrix} =$$

$$= \begin{pmatrix} \frac{1}{1-\beta^2} & \frac{\beta}{1-\beta^2} & 0 \\ \frac{\beta}{1-\beta^2} & \frac{1}{1-\beta^2} & 0 \\ 0 & 0 & \frac{1}{1-\beta} \end{pmatrix} \begin{pmatrix} r_1 \\ 0 \\ r_3 \end{pmatrix} = \begin{pmatrix} \frac{r_1}{1-\beta^2} \\ \frac{r_1\beta}{1-\beta^2} \\ \frac{r_3}{1-\beta^2} \end{pmatrix}$$

$$slope = \nu_3 = \frac{\mathbb{R}_3^{\{1\}\cup\{3\}} - \mathbb{R}_3^{\{1\}}}{\mathbb{W}_3^{\{1\}\cup\{3\}} - \mathbb{W}_3^{\{1\}}} = \frac{\frac{r_3}{1-\beta^2} - \frac{\beta^2 r_1}{1-\beta^2}}{\frac{3}{1-\beta^2} - \frac{\beta^2}{1-\beta^2}} = \frac{r_3 - \beta^2 r_1}{3(1-\beta^2)}$$

In this point, we are explaining that the higher slope is changing depending on the value of α. We define parameter $alpha_{i,j}$ as the value of α for which ν_i=ν_j. In this moment we are interested on $\alpha_{2,3}$.

We know that if $\alpha = 0$ and if $\alpha = 1$ the slopes are equal for all $\beta \in (0,1)$. As we are only considering positive values of α, we will analyze the behaviour of $\nu_2 - \nu_3$ for $\alpha \in [0, \infty)$.

Numerically checking we conclude that if α is smaller than 1, the highest slope is ν_2 for all $\beta \in (0,1)$, but if $\alpha > 1$, then slope ν_2 is less than ν_3. We can observe in Figure 3.4 the following clasif.

(i) If $\alpha < 1$, the highest slope is ν_2. We continue with the Indexing Algorithm defining $\mathcal{S} := \{1,2\}$ and calculating ν_3 and $\mathbb{R}_3^{\{1,2,3\}}$ and $\mathbb{W}_3^{\{1,2,3\}}$.

(ii) If $\alpha > 1$, the highest slope is ν_3 and the algorithm continues defining $\mathcal{S} := \{1,3\}$ and calculating ν_2 and $\mathbb{R}_2^{\{1,2,3\}}$ and $\mathbb{W}_2^{\{1,2,3\}}$.

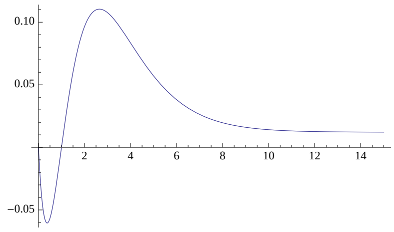


Figure 3.3: Representation of $\nu_3 - \nu_2$

As in both cases, we have to compute $\mathbb{R}^{\mathcal{N}}$ and $\mathbb{W}^{\mathcal{N}}$ we get them as follows:

$$\mathbb{W}^{\mathcal{N}} = \begin{pmatrix} \mathbb{W}_1 \\ \mathbb{W}_2 \\ \mathbb{W}_3 \end{pmatrix} = \left[\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} - \beta \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix} \right]^{-1} \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} = \begin{pmatrix} 1 & -\beta & 0 \\ 0 & 1 & -\beta \\ 0 & 0 & 1-\beta \end{pmatrix}^{-1} \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} =$$

$$= \begin{pmatrix} 1 & \beta & \frac{\beta^2}{1-\beta} \\ 0 & 1 & \frac{\beta}{1-\beta} \\ 0 & 0 & \frac{1}{1-\beta} \end{pmatrix} \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} = \begin{pmatrix} \frac{1+\beta+\beta^2}{1-\beta} \\ \frac{2+\beta}{1-\beta} \\ \frac{3}{1-\beta} \end{pmatrix}$$

$$\mathbb{R}^{\mathcal{N}} = \begin{pmatrix} \mathbb{R}_1 \\ \mathbb{R}_2 \\ \mathbb{R}_3 \end{pmatrix} = \left[\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} - \beta \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix} \right]^{-1} \begin{pmatrix} r_1 \\ r_2 \\ r_3 \end{pmatrix} = \begin{pmatrix} 1 & -\beta & 0 \\ 0 & 1 & -\beta \\ 0 & 0 & 1-\beta \end{pmatrix}^{-1} \begin{pmatrix} r_1 \\ r_2 \\ r_3 \end{pmatrix} =$$

$$= \begin{pmatrix} 1 & \beta & \frac{\beta^2}{1-\beta} \\ 0 & 1 & \frac{\beta}{1-\beta} \\ 0 & 0 & \frac{1}{1-\beta} \end{pmatrix} \begin{pmatrix} r_1 \\ r_2 \\ r_3 \end{pmatrix} = \begin{pmatrix} r_1 + \frac{\beta r_2}{1-\beta} + \frac{\beta^2 r_3}{1-\beta} \\ r_2 + \frac{\beta r_3}{1-\beta} \\ \frac{r_3}{1-\beta} \end{pmatrix}$$

Now, we calculate the slopes in each instance:

(i) If $\alpha < 1$,

$$slope = \nu_3 = \frac{\mathbb{R}_3^{\{1,2\}\cup\{3\}} - \mathbb{R}_3^{\{1,2\}}}{\mathbb{W}_3^{\{1,2\}\cup\{3\}} - \mathbb{W}_3^{\{1,2\}}} = \frac{\frac{r_3}{1-\beta} - \frac{\beta r_2}{1-\beta^2}}{\frac{3}{1-\beta} + \frac{2\beta}{1-\beta^2}} = \frac{r_3 + \beta(r_3 - r_2)}{3 + \beta(3-2)}$$

(ii) If $\alpha > 1$

$$slope = \nu_2 = \frac{\mathbb{R}_2^{\{1,3\}\cup\{2\}} - \mathbb{R}_2^{\{1,3\}}}{\mathbb{W}_2^{\{1,3\}\cup\{2\}} - \mathbb{W}_2^{\{1,3\}}} = \frac{r_2 + \frac{\beta r_3}{1-\beta} - \frac{\beta r_1}{1-\beta^2}}{\frac{2+\beta}{1-\beta} - \frac{\beta}{1-\beta^2}} = \frac{r_2 + \beta(r_3 - r_1) + \beta^2(r_3 - r_2)}{2 + 2\beta + \beta^2} =$$

$$= \frac{r_2 + \beta(r_3 - r_1) + \beta^2(r_3 - r_2)}{2 + \beta(3-1) + \beta^2(3-2)}.$$

Now, we are describing if appeared two problems are indexable and we will test conditions for indexability in each instance.

For $\alpha < 1$, we have to prove that $\nu_1 \geq \nu_2 \geq \nu_3$ to assume indexability in this case. We prove $\nu_1 \geq \nu_2$ as follows:

$$\nu_2 = \frac{r_2 - \beta r_1}{2-\beta} \leq \frac{2r_1 - \beta r_1}{2-\beta} = r_1 = \nu_1$$

In the inequality we are using that in Figure 3.2 we can observe that $\frac{r_3}{3} \leq \frac{r_2}{2} \leq r_1$ for all positive α.

We are going to prove know that $\nu_2 \geq \nu_3$ for all $\beta \in (0,1)$.

$$\nu_2 = \frac{r_2 - \beta r_1}{2-\beta} \geq \frac{\beta r_2 - r_3(1+\beta)}{3+\beta} = \nu_3$$

If we rearrange both sides of the inequality we get the following expression.

$$r_1(\beta^2 + 3\beta) + r_2(\beta^2 - 3\beta - 3) + r_3(-\beta^2 + \beta + 2) \leq 0$$

As we are considering values of $\beta \in (0,1)$, we can claim that the left side of the expression is less than $4r_1 - 5r_2 + 2r_3$.

In the first step we concluded that $r_1 \geq \frac{r_2}{2} \geq \frac{r_3}{3}$ and therefore, we can say that always happens the following:

$4r_1 - 5r_2 + 2r_3 \leq 4r_1 - 10r_2 + 6r_1 = 0$

So, that implies that $\nu_2 \geq \nu_3$ for all β. According to what we have just done, we can claim the following theorem:

Theorem 3.3. *Three states TCP model with additive increase and non-restating decrease if $\alpha < 1$ is indexable because the slopes obtained are non-increasing ($\nu_1 \geq \nu_2 \geq \nu_3$) and can be solved under threshold policy due to the order of appearance of the states$(1, 2, 3)$ and the index values obtained are the following:*

- $\nu_1 = r_1$
- $\nu_2 = \frac{r_2 - \beta r_1}{2-\beta}$
- $\nu_3 = \frac{r_3 + \beta(r_3 - r_2)}{3 + \beta(3-2)}$

For $\alpha > 1$, the order of best states appeared is not increasing, so that means that the problem can not be solved under threshold policy, but we are going to prove that the problem is indexable. What we have to demonstrate is the following: $\nu_1 \geq \nu_3 \geq_2$. The first inequality is proved as follows:

$$\nu_3 = \frac{r_3 - \beta^2 r_1}{3 - \beta^2} \leq \frac{3r_1 - \beta^2 r_1}{3 - \beta^2} = r_1 = \nu_1$$

In the inequality we are using the inequality that we can see in Figure 3.2 ($\frac{r_3}{3} \leq \frac{r_2}{2} \leq r_1$).

Now, we are proving that $\nu_3 \geq \nu_2$ in the following manner:

$$\nu_3 = \frac{r_3 - \beta^2 r_1}{3 - \beta^2} \geq \frac{r_2 + \beta(r_3 - r_1) + \beta^2(r_3 - r_2)}{2 + 2\beta + \beta^2} = \nu_2$$

.

We if rearrange the expression above, we obtain the following formula:

$$r_1(\beta^4 + 5\beta^3 - 2\beta^2 - 3\beta) + r_2(\beta^4 - 4\beta^2 + 3) + r_3(-\beta^4 - \beta^3 + 2\beta^2 + \beta - 2) \leq 0$$

As we know that $\frac{r_3}{3} \leq \frac{r_2}{2} \leq r_1$ and $\beta \in (0,1)$ the left side of the expression is less than 0, operating in the same manner as we did before. So, the inequality we are proving always holds.

According to all that we have just prove we can claim the following theorem:

Theorem 3.4. *Three states TCP model with additive increase and non-restating decrease, if $alpha > 1$ then is indexable because the slopes obtained are non-increasing ($\nu_1 \geq \nu_3 \geq \nu_2$). For all ν, there is an optimal policy which is either $\emptyset$ or $\{1\}$ or $\{1,3\}$ or $\{1,2,3\}$ and the index values are:*

- $\nu_1 = r_1$
- $\nu_3 = \frac{r_3 - \beta^2 r_1}{3 - \beta^2}$
- $\nu_2 = \frac{r_2 + \beta(r_3 - r_1) + \beta^2(r_3 - r_2)}{2 + (3-1)\beta + (3-2)\beta^2}$

As we said before, in $\alpha = 1$ instance, the highest slope computed in the first step is ν_1. Now, we are getting the behaviour of this critical value of α. As we know the values of the slopes from what we did before, we are going to check what is the highest slope in the second step. It can be observed in Figure 3.4, for all $\beta \in (0,1)$ that $\nu_3 - \nu_2$ is always negative, so we can conclude that the highest slope is ν_3.

After this step, we know that we last index is ν_2 and we can conclude that this is an special case of $\alpha > 1$ instance. So, we claim the following colorary:

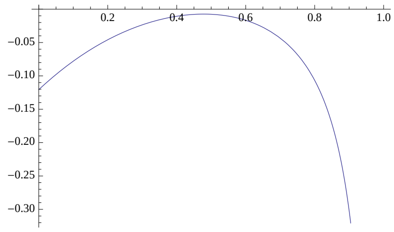


Figure 3.4: $\nu_3 - \nu_2$ for $\alpha = 1$

Corollary 3.1. *Three states TCP model with additive increase and non-restating decrease, if $\alpha = 1$ then we assume that the problem is indexable because the slopes obtained are non-increasing ($\nu_1 \geq \nu_3 \geq \nu_2$). If the problem is indexable, then for all ν, there is an optimal policy which is either $\emptyset$ or $\{1\}$ or $\{1,3\}$ or $\{1,2,3\}$ and the index values are:*

- $\nu_1 = r_1$
- $\nu_3 = \frac{\frac{\log 4}{3} - \beta^2 \log 2}{3-\beta^2}$
- $\nu_2 = \frac{\frac{\log 3}{2} + \beta(\frac{\log 4}{3} - log 2) + \beta^2(\frac{log 4}{3} - \frac{\log 3}{2})}{2+(3-1)\beta+(3-2)\beta^2}$

To sum up, we are including Figure 3.5, that represents the output of the algorithm under the numerical process that Matlab realizes and we are going to conclude that the output is the same.

In this numerical algorithm all the matrices and vectors are defined in the same way as in the theorical prove we have just done and the discount factor used is 0.999. It is not possible to know if the slopes computed above are the same as in the figure just observing the picture, but we can distinguish the behaviour when $\alpha < 1$ and when $\alpha > 1$. In the case that the problem can be solved under threshold policy ($\alpha = 0$, $\alpha = 0.5$) we observe three points and in other cases we observe only two points. The obtained sequence is $\mathcal{S} = \{1,3,2\}$ in case that $\alpha > 1$ and that implies modifications in figure the figure that do not let us see all the points. So, we can conclude from the figure obtained of the algorithm that the problem can not solved under threshold policy if we observe less point that we should appear.

On the other hand, we claim that just observing the figure, we can conclude whether it is possible assuming indexability or not, because it is very easy to see if the slopes obtained are non increasing or not in the picture.

3.5.2 Numerical Computing of Index Values

Programming in Matlab the described algorithm is the first step in the work that we have done. We know that the restarting case is indexable Jacko and Sanso (2010), but

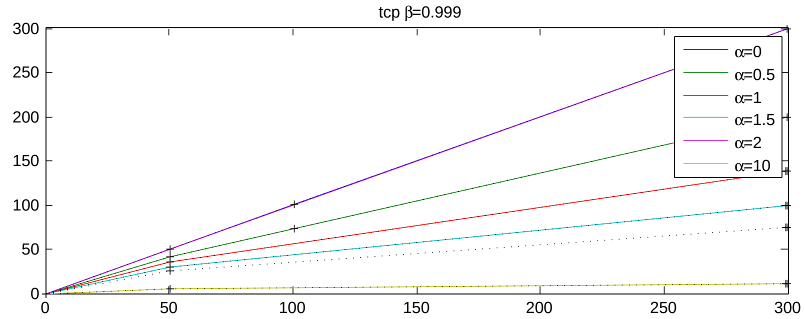


Figure 3.5: Output of Checking Indexing Algorithm for three states TCP model

we run the algorithm for different decreasing factor and we conclude if we see the indexability in all cases and if it is possible to solve the optimization problem using a threshold policy.

The discount factor that is going to be used in our first results in Matlab is 0.999. If $\beta = 1$ the problem will be good modeled, but in this case, $I - \beta \boldsymbol{P}$ would not be regular and the algorithm is not applicable in this instance. (Remember that $\boldsymbol{P}$ is the matrix that consist on $p_{n,m}^{a(t)|\mathcal{S}}$ for all $m, n \in \mathcal{N}$, where $p_{n,m}^{a(t)|\mathcal{S}}$ is the probability of going from state n to state m if decision $a(t)$ is decided under policy $\mathcal{S} \subset \mathcal{N}$.)

We will show the behaviour of five states and twenty-five states TCP model with additive increasing and the different decrease factor.

We will be interest also on studying the model for a decrease factor that consist on transmitting one packet less than in the previous time slot if decided action is 0. This case is called *birth-death instance* because if action 1 is decided it is transmitted one packet more and if action 0 is decided one packet less is transmitted.

We will be able to represent *birth-death instance* using a decrease factor $1 - \delta$, where δ is small enough.

In all the instances that we are going to present the algorithm used is the *Checking Indexing Algorithm* described before. Thus, we going to use what we explained in that section to conclude indexability in our model.

Five states model

First, we will observe the behaviour of the five states TCP model for additive increase and all possible decreasing that in five state model is possible. We consider that a decreasing is different for two given values of γ if the $\boldsymbol{P}^0$ matrix computed in both cases is not equal.

First, we are describing all possible $\boldsymbol{P}^0$ matrices that decrease factor can create:

(i) If $\gamma \in [0, \frac{2}{5})$, the matrix $\boldsymbol{P}^0$ is $\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix}$

(ii) If $\gamma \in [\frac{2}{5}, \frac{1}{2})$, the matrix $\boldsymbol{P}^0$ is $\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix}$

(iii) If $\gamma \in [\frac{1}{2}, \frac{3}{5})$, the matrix $\boldsymbol{P}^0$ is $\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix}$

(iv) If $\gamma \in [\frac{3}{5}, \frac{2}{3})$, the matrix $\boldsymbol{P}^0$ is $\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix}$

(v) If $\gamma \in [\frac{2}{3}, \frac{3}{4})$, the matrix $\boldsymbol{P}^0$ is $\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix}$

(vi) If $\gamma \in [\frac{3}{4}, \frac{4}{5})$, the matrix $\boldsymbol{P}^0$ is $\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix}$

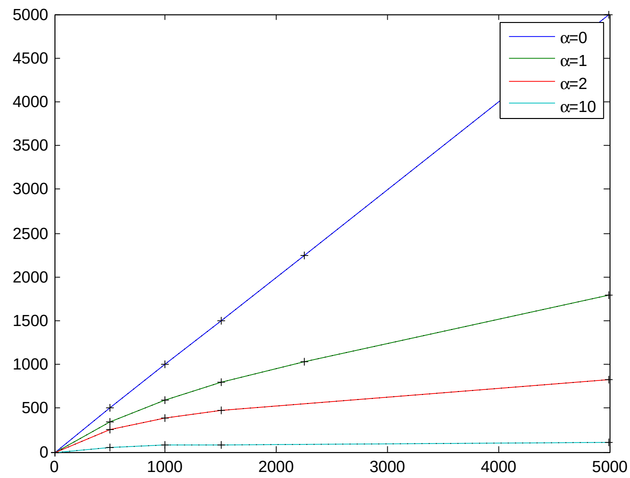


Figure 3.6: Output of algorithm of 5 states model with additive increase and $\gamma \in [\frac{2}{5}, \frac{1}{2})$

(vii) If $\gamma \in [\frac{4}{5}, 1)$, the matrix $\boldsymbol{P}^0$ is $\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}$

In the first case ($\gamma \in [0, \frac{2}{5})$), we are describing the restarting case that we know from Jacko and Sanso (2010) it is indexable problem, that can be solved under thrshold policy and how we can obtain the indices. So, that is why we are not considering this instance.

We will show the output of the algorithm for different α-s (0, 1, 2 and 10) and in two different figures for the other cases. In the first first, we can see the output of the algorithm as the solution of problem (2.3). In the second figure it is represented the states of $\mathcal{N}$ in x-axis and the slope connected to each state in y-axis. In this way, we can see in a better way how the slopes are in our model and the order of appearance of the states.

In Figure 3.6 we can observe the output of the Checking Indexing Algorithm as a solution of the defined maximum problem for a decrease factor $\gamma \in [\frac{2}{5}, \frac{1}{2})$ and in Figure 3.7 we can observe the slopes and the order of appearance of the states.

In both figures we can see that the slopes are non-increasing, so we can assume indexability. But, as *PCL*-indexability is true, we can claim that the problem is indexable.

Moreover, we can observe the order of appearance of the states in the Algorithm is $1, 2, 3, 4, 5$ if the value of $\alpha \leq 1$. That implies that problem can be solved under threshold policy if satisfies $\alpha \leq 1$.

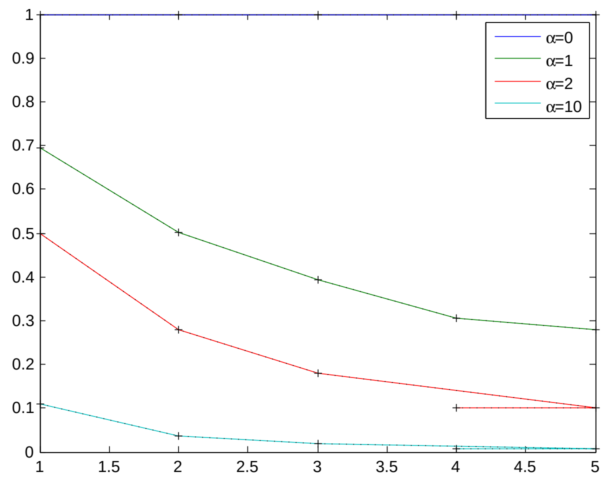


Figure 3.7: Indices of 5 states TCP model with additive increase and $\gamma \in [\frac{2}{5}, \frac{1}{2})$

In $\alpha = 2$ and $\alpha = 10$ instances, the order of appearance of states is $1, 2, 3, 5, 4$. Therefore, since the problem is indexable, then for all ν, there is an optimal policy which is either $\{\emptyset\}$ or $\{1\}$ or $\{1, 2\}$ or $\{1, 2, 3\}$ or $\{1, 2, 3, 5\}$ or $\{1, 2, 3, 4, 5\}$.

Theorem 3.5. *Five states TCP model, if we consider additive increase and decrease factor $\gamma \in [\frac{2}{5}, \frac{1}{2})$, is always indexable and can be solved under threshold policy if $\alpha \leq 1$. In contrast, if $\alpha > 1$, for all ν, there is an optimal policy which is either$\{\emptyset\}$ or $\{1\}$ or $\{1, 2\}$ or $\{1, 2, 3\}$ or $\{1, 2, 3, 5\}$ or $\{1, 2, 3, 4, 5\}$.*

In Figure 3.8 and in Figure 3.9 we can see the behaviour of the model for a decrease factor $\gamma \in [\frac{1}{2}, \frac{3}{5})$. It can be observed in this representations that the slopes are non-increasing, so we can assume indexability. Besides, the order of appearance of the states in the Algorithm is $1, 2, 3, 4, 5$ if the value of $\alpha \leq 1$. That implies that problem can be solved under threshold policy if satisfies $\alpha \leq 1$.

In $\alpha = 2$ instance, the order of appearance of states is $1, 2, 4, 3, 5$ and if $\alpha = 10$ the order is $1, 2, 5, 4, 3$. That means that the problem can not be solved under threshold policy if $\alpha > 1$.

The check related to *PCL*-indexability let us conclude that the problem is indexable, although some instances can not be solved under threshold policy.

Theorem 3.6. *Five states TCP model, if we consider additive increase and decrease factor $\gamma \in [\frac{1}{2}, \frac{3}{5})$, is always indexable and can be solved under threshold policy if $\alpha \leq 1$. In contrast, if $\alpha > 1$, threshold policy may not be optimal.*

If we consider a decrease factor $\gamma \in [\frac{3}{5}, \frac{2}{3})$, we can observe the behavior of our model in Figure 3.10 and Figure 3.11. As we can be shown in the figures, the slopes are non-increasing and that means that we can assume indexability.

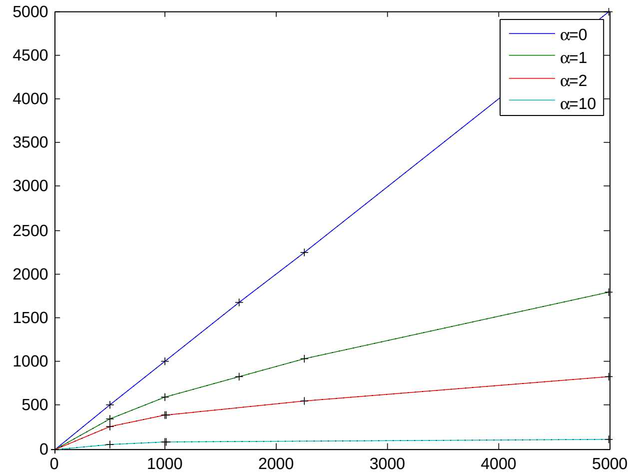


Figure 3.8: Ouput of algorithm of 5 states TCP model with additive increase and $\gamma \in [\frac{1}{2}, \frac{3}{5})$

We are interested on the order of appearance of the states in our algorithm to know if threshold policy can be applied to solve the problem (2.3). The figures tell us that only $\alpha = 0$ can be solved under threshold policy.

In $\alpha = 1$ and $\alpha = 2$, the order of appearance of states is $1, 2, 3, 5, 4$, so if the problem is indexable, then for all ν, there is an optimal policy which is either $\emptyset$, 1 or $1, 2$ or $1, 2, 3$ of $1, 2, 3, 5$ of $1, 2, 3, 4, 5$. But if $\alpha = 10$, the order is as follows: $\mathcal{S} = 1, 2, 5, 4, 3$, therefore for all ν, there is an optimal policy which is either $\emptyset$ or 1 or $1, 2$ or $1, 2, 5$ or $1, 2, 4, 5$ or $1, 2, 3, 4, 5$.

The *PCL*-indexability check let us claim that the problem is indexable.

Theorem 3.7. *Five states TCP model, if we consider additive increase and decrease factor $\gamma \in [\frac{3}{5}, \frac{2}{3})$, is always indexable and can be solved under threshold policy if $\alpha = 0$. However, if $\alpha \neq 0$, threshold policy may not be optimal.*

We are analyzing the instance of decrease factor $\gamma \in [\frac{2}{3}, \frac{3}{4})$. The matrix that this γ creates is defined above and in Figure 3.12 and Figure 3.13 we can understand the behaviour of the model. As we can observe in the figures, the slopes are non-increasing and that means that we can assume indexabiity.

We are interested on the order of appearance of the states in our algorithm to know if threshold policy can be applied to solve problem (2.3). Only if $\alpha = 0$ the problem can be solved under this threshold policy.

According to the *PCL*-indexability check, the problem is indexable.

Theorem 3.8. *Five states TCP model, if we consider additive increase and decrease factor $\gamma \in [\frac{2}{3}, \frac{3}{4})$, is always indexable and can be solved under threshold policy if $\alpha = 0$. However, if $\alpha \neq 0$,*

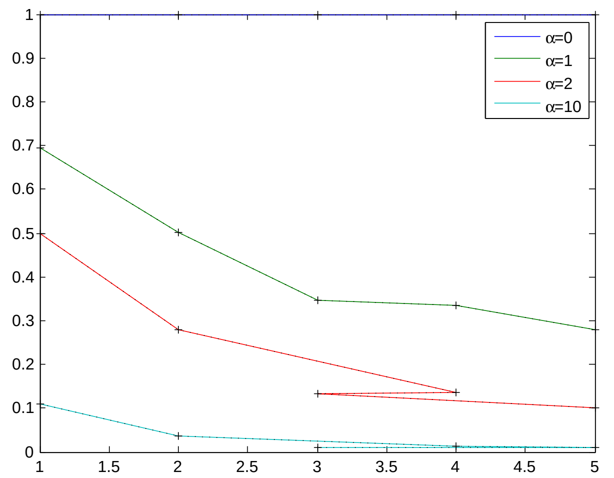


Figure 3.9: Indices of 5 states TCP model with additive increase and $\gamma \in [\frac{1}{2}, \frac{3}{5})$

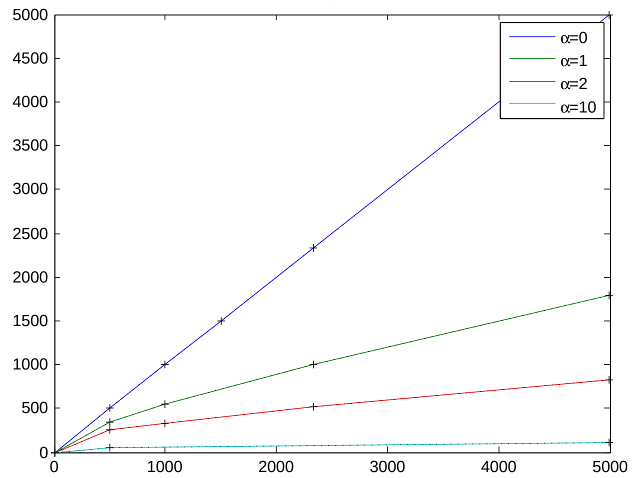


Figure 3.10: Ouput of algorithm of 5 states TCP model for $\gamma \in [\frac{3}{5}, \frac{3}{4})$

for all ν, threshold policy may not be optimal.

The decrease factor value $\gamma \in [\frac{3}{4}, \frac{4}{5})$ return different $\boldsymbol{P}^0$ matrix comparing with the other values of γ we showed until this moment. So, we are seeing what is happening in this instance in Figure 3.14 and Figure 3.15.

In the figures that represent this instance, we can see that the slopes are non-increasing, so we can assume indexability in this instance too. Besides, the problem is *PCL*-indexable (*PCL*-indexability is true) so we claim that the problem is indexable.

According to the order of appearance of states, only $\alpha = 0$ case can be solved under threshold policy.

Theorem 3.9. *Five states TCP model, if we consider additive increase and decrease factor $\gamma \in$*

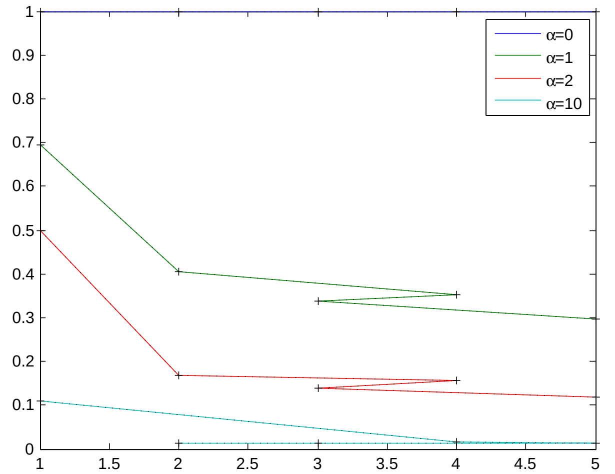


Figure 3.11: Indices of 5 states TCP model for $\gamma \in [\frac{3}{5}, \frac{3}{4})$

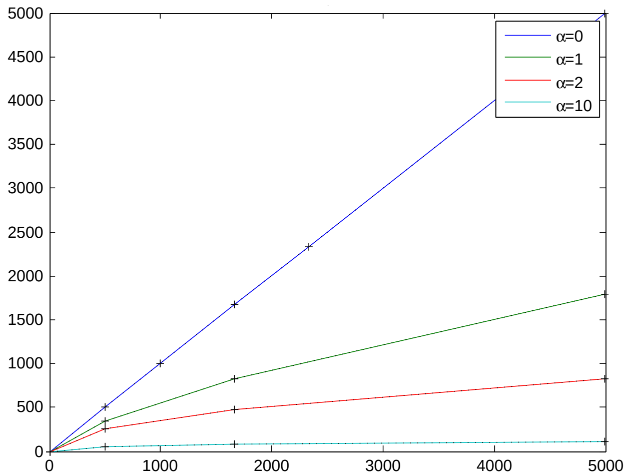


Figure 3.12: Ouput of algorithm of 5 states TCP model for $\gamma \in [\frac{2}{3}, \frac{3}{4})$

$[\frac{3}{4}, \frac{4}{5})$, is always indexable and can be solved under threshold policy if $\alpha = 0$. However, if $\alpha \neq 0$, threshold policy may not be optimal.

The last instance is connected to the birth-death instance ($\gamma \in [\frac{4}{5}, 1)$) for a five states model. In this case, increasing is the same as in the previous examples, but the decreasing is one by one. The results obtained for this model are represented in Figure 3.16 and Figure 3.17.

As we can observe in Figure 3.17 the slopes resulted are always non-increasing for all $\alpha - s$. That means that we can assume that the problem is indexable. However, the $\mathcal{S}$ set obtained in the algorithm is $\{1, 2, 3, 4, 5\}$ only if $\alpha = 0$, so the problem can be solved under threshold policy if $\alpha = 0$.

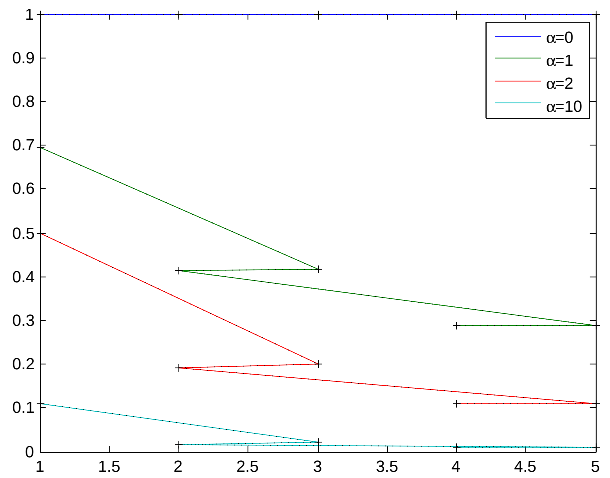


Figure 3.13: Indices of 5 states TCP model for $\gamma \in [\frac{2}{3}, \frac{3}{4})$

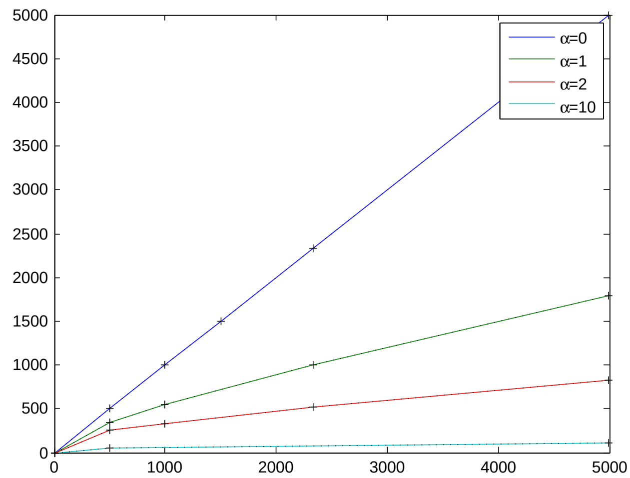


Figure 3.14: Ouput of algorithm of 5 states TCP model for $\gamma \in [\frac{3}{4}, \frac{4}{5})$

The *PCL*-indexability check ensures that the problem is indexable, althought some instances can not be solved under threshold policy.

Theorem 3.10. *For five states TCP model, birth-death instance for a five states TCP model is always indexable and can be solved under threshold policy if* $\alpha = 0$.

Twenty-five states model

We will continue analyzing the same increasing and decreasing TCP models as we have carried out for five states model.

We will also try to understand the behaviour of this model using two pictures: one that shows the solution of problem (2.3) and in the other figure can be observed the

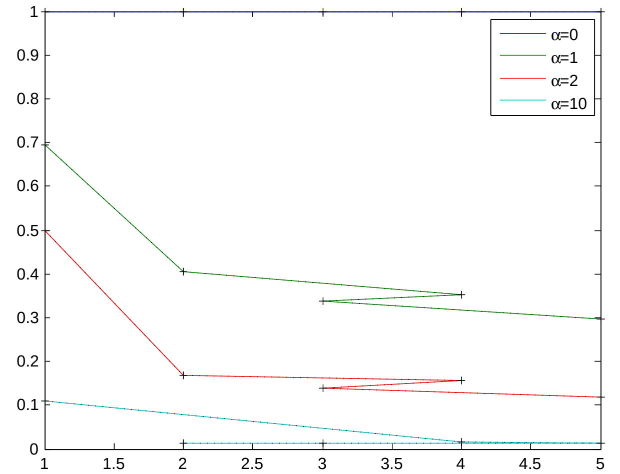


Figure 3.15: Indices of 5 states TCP model for $\gamma \in [\frac{3}{4}, \frac{4}{5})$

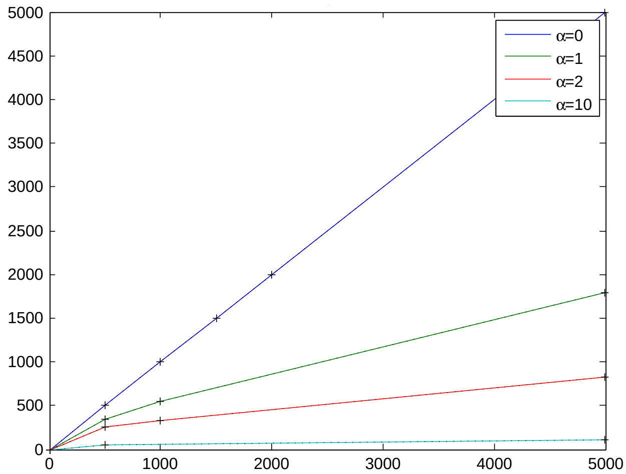


Figure 3.16: Ouput of algorithm of 5 states birth-death TCP model

indices obtained in the algorithm.

For all the figures we are going to show the values of α that we compute are the same as in the five states example. That will let us claim the differences between those two models.

In case we are considering a decrease factor of a third, Figure 3.18 and Figure 3.19 show the behavior of our model.

We can assume that the problem is indexable because the slopes calculated are non-increasing. in addition, the order of appearance is $1, 2, 3, ..., 24, 25$ if $\alpha \leq 1$, so the problem can be solved under threshold policy only in this case.

We can not conclude the problem is indexable yet. But, as *PCL*-indexability is true, we can claim that the problem is indexable.

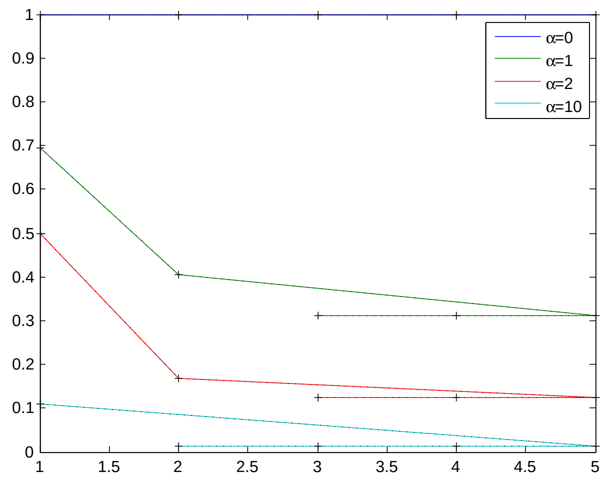


Figure 3.17: Indices of 5 states birth-death TCP model

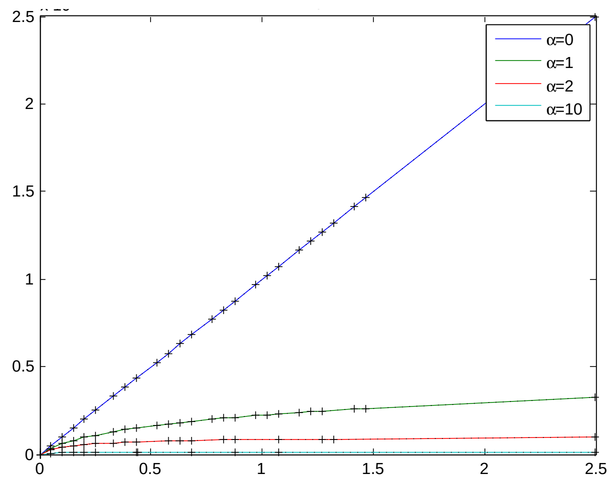


Figure 3.18: Output of algorithm of 25 states model with additive increase and $\gamma = 0.3$

Theorem 3.11. *Twenty-five states TCP model, if we consider additive increase and decrease factor of a third, is always indexable and can be solved under threshold policy only if* $\alpha \leq 1$*.*

Now, we will show a representation of the solution of the algorithm in Figure 3.20 obtained in a twenty-five states TCP model with additive increase and decreasing factor a half.

In Figure 3.20 and Figure 3.21 we can see the new representation of the twenty-five states TCP model for decrease factor of a half. As it can be observed in this figures, the slopes are always non-increasing for all the α-s we take into account, so that is why we can assume indexability. Besides, the order of appearance the states is $1, 2, 3, ..., 23, 24, 25$ only if $\alpha = 0$, so that let us claim that the problem can be solved

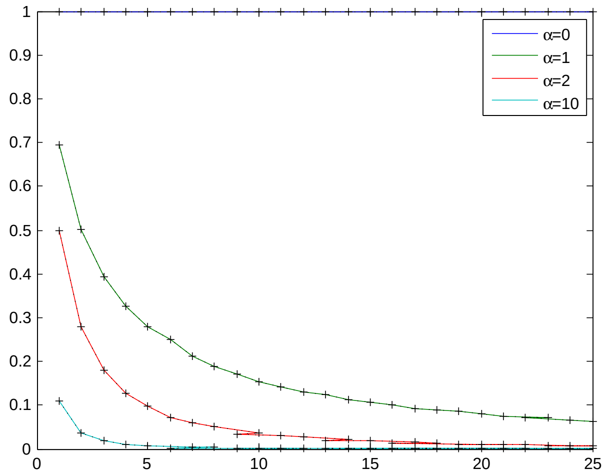


Figure 3.19: Indices of 25 states TCP model with additive increase and $\gamma = \frac{1}{3}$

under threshold policy if $\alpha = 0$. If $\alpha \neq 0$, the order of appearance of states are the following:

- If $\alpha = 1$, we obtain this order of states:

$$1, 2, 3, 4, 6, 5, 8, 7, 10, 9, 12, 11, 14, 13, 16, 15, 18, 17, 20, 19, 22, 24, 21, 25, 23$$

- If $\alpha = 2$, the order of states is:

$$1, 2, 4, 3, 6, 5, 8, 7, 10, 12, 9, 14, 11, 16, 13, 18, 20, 15, 22, 17, 24, 25, 23, 21, 19$$

- If $\alpha = 10$, we get the following order of states:

$$1, 2, 4, 3, 6, 8, 5, 10, 7, 12, 14, 9, 16, 11, 18, 20, 13, 22, 15, 24, 25, 23, 21, 19, 17$$

The changes in the order of appearance of the states in the algorithm is greater if the parameter α is higher.

The check related to *PCL*-indexability tell us that this problem is indexable, although only $\alpha = 0$ instance can be solved under threshold policy.

Theorem 3.12. *Five states TCP model, if we consider additive increase and decrease factor of a half, is always indexable and can be solved under threshold policy if* $\alpha = 0$.

We analyze birth-death instance for a twenty-five states TCP model. In this case, increasing is the same as in the previous examples, but the decreasing is one by one. The results obtained for this model are represented in Figure 3.22 and Figure 3.23.

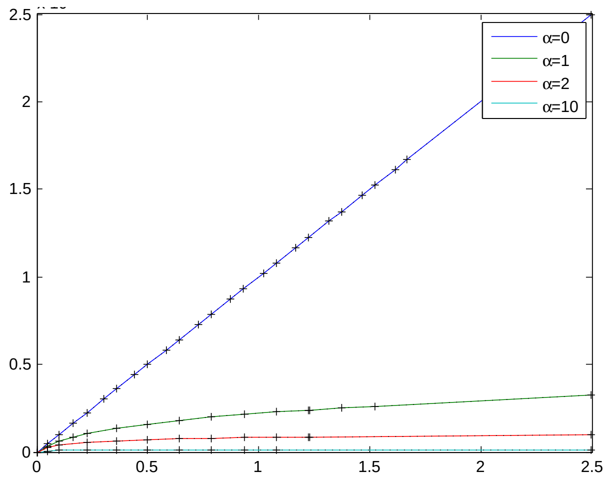


Figure 3.20: Ouput of algorithm of 25 states TCP model with additive increase and $\gamma = 0.5$

As we can observe that the slopes resulted are always non-increasing for all $\alpha - s$. That means that we assume the problem is indexable. However, the $\mathcal{S}$ set obtained in the algorithm is $\{1, 2, 3, ..., 23, 24, 25\}$ only if $\alpha = 0$, so the problem can be solved under threshold policy if $\alpha = 0$.If $\alpha \neq 0$, the order of appearance of states is as follows:

- If $\alpha = 1$, we obtain this order of states:

$$1, 2, 3, 4, 5, 6, 7, 8, 9, 25, 24, 23, 22, 21, 20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10$$

- If $\alpha = 2$, the order of states is:

$$1, 2, 3, 4, 5, 25, 6, 24, 23, 22, 21, 20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7$$

- If $\alpha = 2$, we get the following order of states:

$$1, 25, 24, 23, 22, 21, 20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2$$

As it happened in $\gamma = 0.5$ instance, the changes in the order of appearance of the states in the algorithm is also greater if the parameter α is higher.

The *PCL*-indexability check ensures that the problem is indexable, although only $\alpha = 0$ can be solved under threshold policy.

Theorem 3.13. *Birth-death instance for a twenty-five states TCP model is always indexable and can be solved under threshold policy if $\alpha = 0$.*

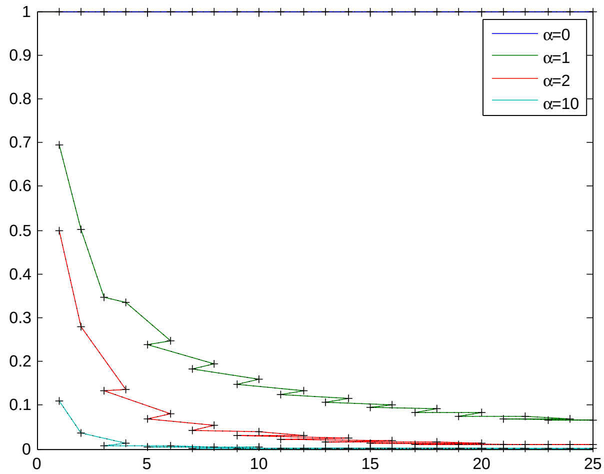


Figure 3.21: Indices of 25 states TCP model with additive increase and $\gamma = 0.5$

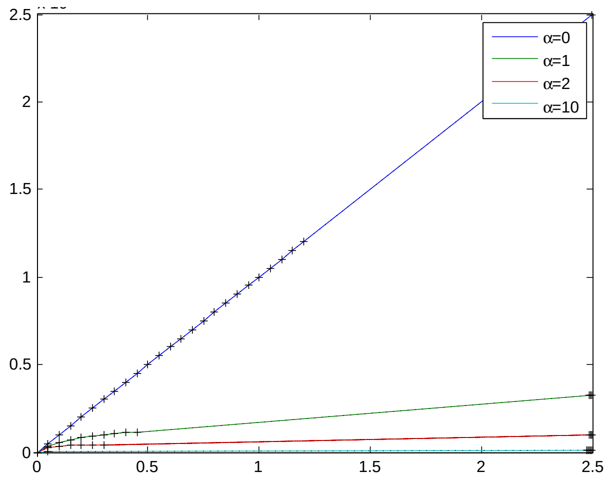


Figure 3.22: Ouput of algorithm of 25 states birth-death TCP model

Comparing numerical results

Now, we are seeing the differences and similarities in the results obtained in twenty-five states model and in five states model.

In both cases, $\alpha = 0$ is indexable and can be solved under threshold policy. That is an obvious result due to the slopes computed are always one, because the reward obtained and number of packets sent is always equal. We call *trivial* to this instance.

The five state TCP model, if we use a decrease factor of a third, is always indexable and it can be solved under threshold policy for all α-s we have computed. Twenty-five states model is always indexable too, but only in case that the value of α is less or equal than one, the problem can be solved under threshol policy.

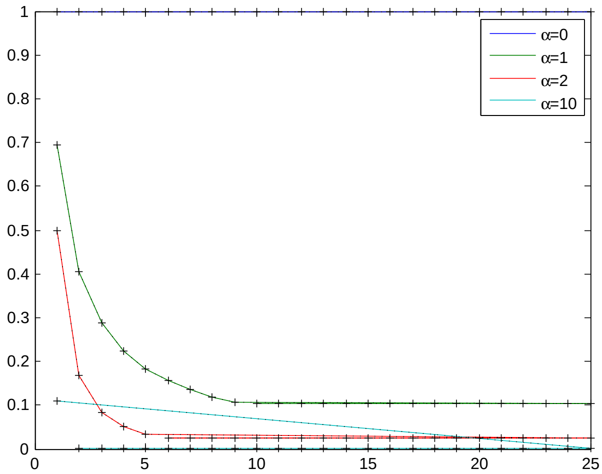


Figure 3.23: Indices of 25 states birth-death TCP model

Using a decrease factor of a third and changing the number of states of our model from five to twenty-five, do not cause changes in indexability, but the problem can be solved under threshold policy in different situations: in five states model satisfies this condition for any $\alpha \leq 1$ and in twenty-five states model only if $\alpha = 0$.

We have just claim that there are some differences if we consider a decrease factor of a third or if we use as a decrease factor the half. Nevertheless, birth-death instance has very similar behaviour in both examples: they are indexable always and they can be solved under threshold policy only if $\alpha = 0$. The changes in the order of states is more complicated in case of twenty-five states model and that implies more difficulty on applying threshold policy.

Chapter 4

Simulation Results

The ns-3 simulator Riley and Henderson (2010) is a discrete-event network simulator targeted primarily for research and educational use.

The ns-3 project, started in 2006, is an open-source project developing ns-3. A few key points are worth noting before continuing:

- Ns-3 is not an extension of ns-2; it is a new simulator. The two simulators are both written in C++ but ns-3 is a new simulator that does not support the ns-2 APIs. Some models from ns-2 have already been ported from ns-2 to ns-3.

- Network Simulator-3 is open-source, and the project strives to maintain an open environment for researchers to contribute and share their software.

4.1 Simulation Scenario Description

In this section, we explain the scenario that we have implement in ns-3 to check that the indices policies are more efficient than the routing that we can observe in real-application.

As we described above, we are interested in fast and fair delivery, which requires avoiding the future congestion congestion in a routing work. So first of all, we need that some users access to a router with a link that can be considered perfect, so that no packets will be dropped in this links.

On the other hand, we assume that the router is connected to a server. This link will be considered also as a perfect connexion.

The velocity of the user-router link should be higher than the router-server link, because we are considering a routing work and that implies that we are interested on how the router decides what packets transmit or not transmit.

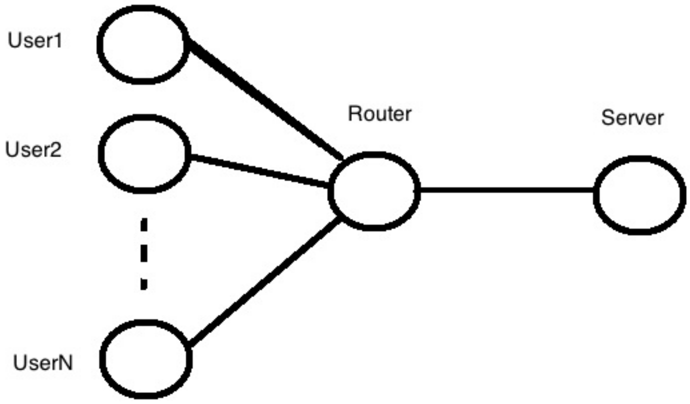


Figure 4.1: Example of simulation scenario for N users

In all our simulations we have got a user-router link of 5Mb/s and a router-server link of a velocity of 1.5Mb/s. We assume that there are some propagation delays in TCP sending packets of all the links of the simulations in order to get more realistic results, 10ms in user-router link and 2ms in router-server link.

To a better understanding, the routing work we are can be observed in Figure 4.1.

As we said before, we are considering that the increasing of the TCP is additive, so that means that we need to define a TCP model with no slow start. Moreover, we are implementing in ns-3 the restarting case. That is why TCP Tahoe has been chosen for this implementations. Other decrease models will be included in future work.

The router has to take the decision of transmitting or not transmitting the flow/packet that the users want to send, so there will appear the main part of our work. The decision consists on comparing the received index of the user with a threshold in the following way: if the index is greater than the threshold parameter of the router the router transmits, but if the index is less than the threshold the packet will be dropped.

To finish with the scenario description, we consider that the users are honest and they send the index information that corresponds to the number of packets they are sending.

4.2 Changing the ns-3 Code

In this part we are detailing all the code that we have changed comparing with a routing implementation of ns-3.

There are three periods in all the simulations of ns-3: defining, running and ending.

In the first part of the simulation, all the variables are defined and initiated. In this part of the code, it will be declared the parameters of each user, so we can get the indices with a C++ code with an error less than 10^{-5}, comparing with the slopes resulted in the Matlab algorithm. The inverse matrix is computed using the implementation of Gauss-Jordan method in the following page: *www.elrincondelc.com*.

To do that, we have added a function *get-slopes* in OnOffApplication::StartApplication() function in OnOffApplication.cc file. This function (get-slopes) takes the α and γ parameters of each user and returns the vector of indices of size 70.

The maximum number of packets in the queue is 50. That is the reason of thinking that the size of the vector of indices is 50. However, since the router transmits packets so fast, it could happen that the user sends more than that amount of packets, so we fix the maximum number of packets in 70 to ensure that there is no access to no defined index.

Each user has an application that consists on sending packets to a server. Therefore, we think it is a great idea storing the vector of indices in the application defined for the user.

In the second part of the simulation, we are interested on passing the indices that corresponds to the number that sends each user to the router. Besides, the router has a threshold parameter to compare with the received packets.

As we said before, the vector of indices of a user are saved in the Application of that user. We add *SetIdx* function in TcpSocketBase::SendPendingData() function in tcp-socket-base.cc file to inject the index to the packet.

The main part of the work is implementing a different way of realizing a routing work. So, the enqueueing of the router to the server is changed. We have implemented this part in Packet::DoEnqueue function of Packet.cc file.

To get the index of the packets that arrived to the queue we have added the GetIdx() function that given a packet returns the index that corresponds to that packet.

The decision of the router consists on comparing the received index with the threshold parameter of the router. That part is realized with an IF function. And when the condition of being the index less than the threshold parameter of the router occurs Drop function is called.

In the ending part of the simulation, there is nothing else to do than using the destructors of the objects of ns-3 to get the memory free. Obviously, the destructors of Network Simulator includes all we have added in the code.

4.3 Results

In this section we are describing the output of the simulations we have just presented.

We will present the behavior of one user and two users routing with index policies and we will compare it with no index model to check this new model is interesting.

In both cases we are representing the adding increase and restarting decrease model. Moreover, we define $\alpha = 2$ for all the users.

Also, different threshold parameter will be simulated and we will observe the differences between them, so that we can obtain conclusions.

One user implementation

In this implementation, the decision is transmitting or not transmitting if the received index from the user is less than the threshold parameter of the router.

No index model consists on an threshold parameter of 0 in the router. First of all, we are describing the changing of the congestion window and the queue of the router versus time in the no index model. After that, we are presenting the congestion window, the index of the packet received and the queue of the router versus time.

In Figure 4.3 we can observe how the queue of the router changes in time and in Figure 4.2 the changing of the congestion window . As we can see, in 12 seconds there is a packet dropping because the number of packets in the queue arrives to the maximum.

The number of packets sent grows in the same way if there is no packet dropping and starts sending one packet. However, the queue of the router starts in zero and in the first seconds the number of packets is no more that one because the router is able to transmit all the packets that the user sends.

In one user model we are using the following values of the threshold parameter of the router: 0.02, 0.01, 0.008 and 0.05. And we will observe differences between them in the number of sent packets, in the number of packets in the queue of the router and in the index of the user.

For a threshold value of 0.02, in Figure 4.4 we can observe that the number of packets sent is less or equal to 14.

In Figure 4.5 it is showed that the number of packets in the queue for a threshold parameter of the router is no more than 4.

The figure that shows the index in time is Figure 4.6 and there we can see that there is a packet dropped when the index reaches to the threshold value of 0.02. that implies, that the number of packets sent by the user in that moment goes to one and the queue get empty.

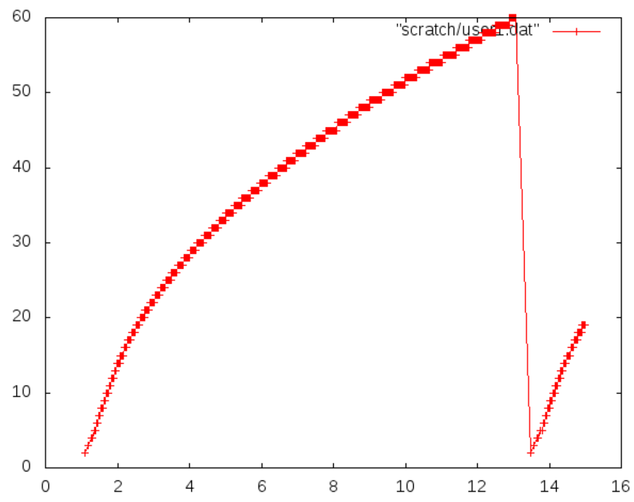


Figure 4.2: The number of packets sent by the user for no index model

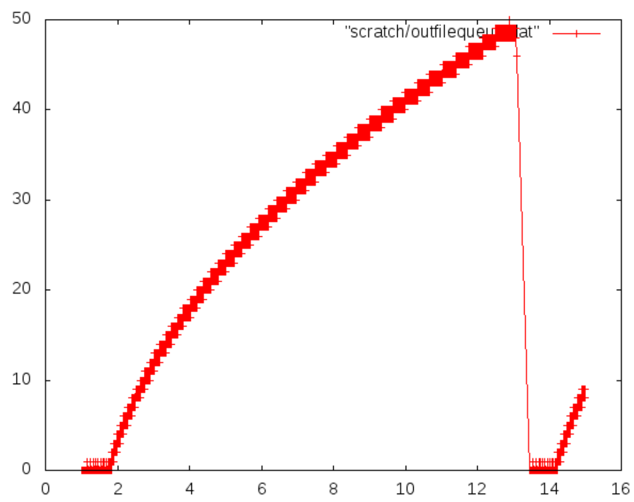


Figure 4.3: The number of packets in the queue of the router in a one user routing for no index model

For a threshold of 0.01, we can observe that in in Figure 4.7 we can observe that the number of packets sent is less or equal to 23.

The number of packets in the queue can be seen in Figure 4.8 and there is no greater value than 13 in the queue.

The figure that shows the index in time is Figure 4.9 and there we can see that there is a packet dropped when the index reaches to the threshold value of 0.01. So, that the number of packets sent by the user in that moment goes to one and the queue get empty.

In this case, the number of packet dropping is less than in the previous instance

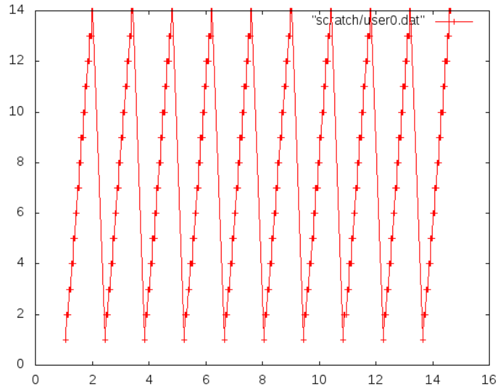


Figure 4.4: The number of packets sent by the user in a one user routing for 0.02 threshold

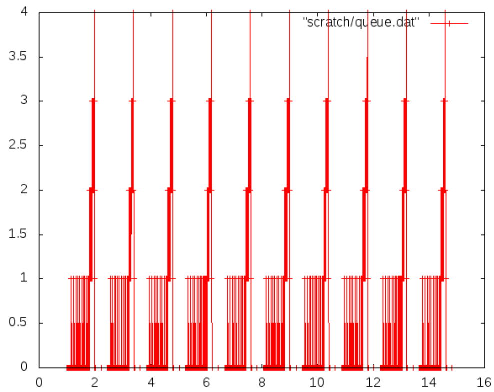


Figure 4.5: The number of packets in the queue of the router in a one user routing for 0.02 threshold

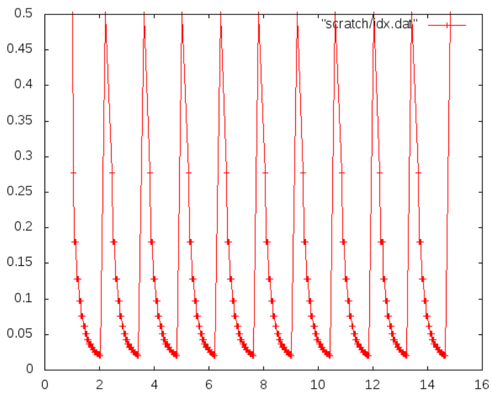


Figure 4.6: The index of the user in a one user routing for 0.02 threshold

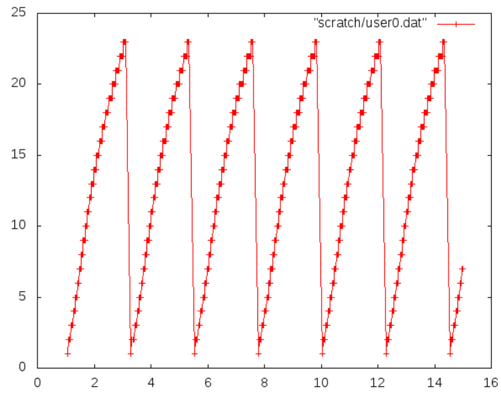


Figure 4.7: The number of packets sent by the user in a one user routing for 0.01 threshold

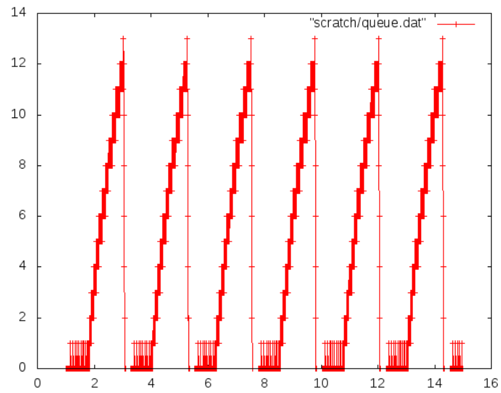


Figure 4.8: The number of packets in the queue of the router in a one user routing for 0.01 threshold

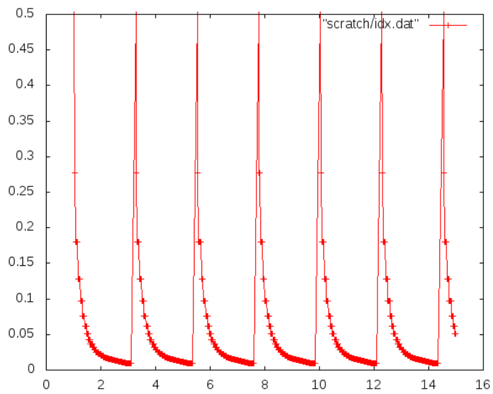


Figure 4.9: The index of the user in a one user routing for 0.01 threshold

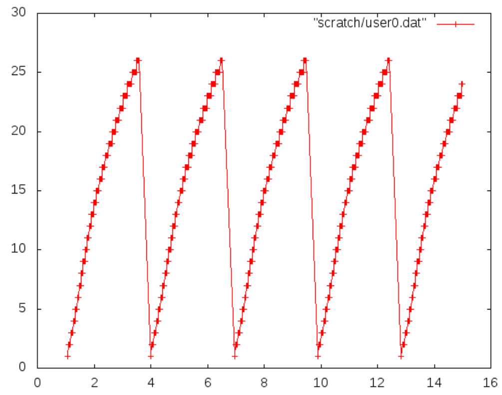


Figure 4.10: The number of packets sent by the user in a one user routing for 0.008 threshold

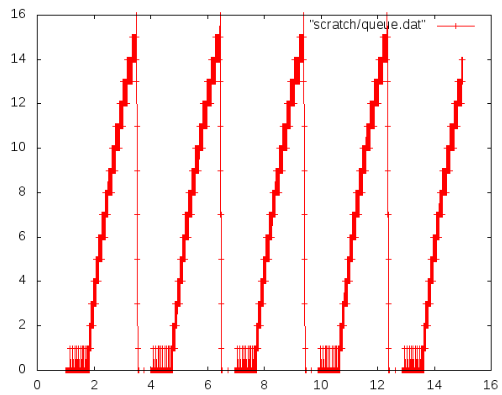


Figure 4.11: The number of packets in the queue of the router in a one user routing for 0.008 threshold

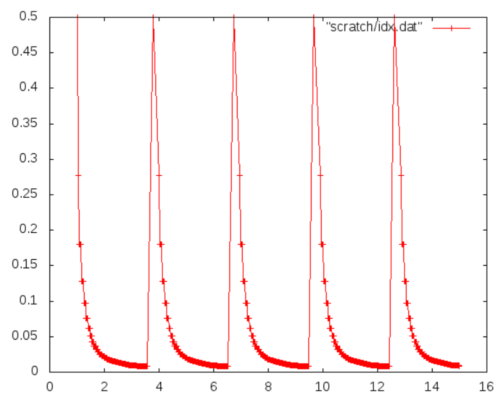


Figure 4.12: The index of the user in a one user routing for 0.008 threshold

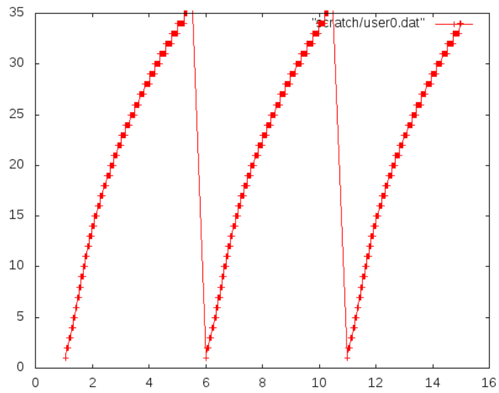


Figure 4.13: The number of packets sent by the user in a one user routing for 0.005 threshold

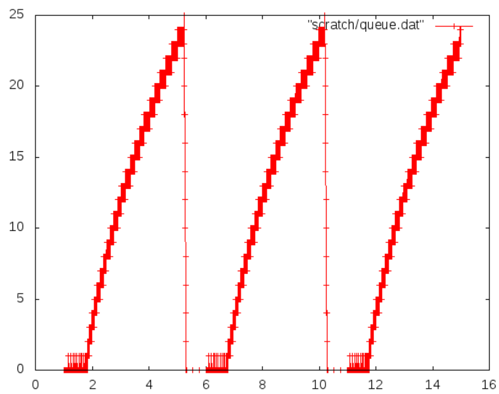


Figure 4.14: The number of packets in the queue of the router in a one user routing for 0.005 threshold

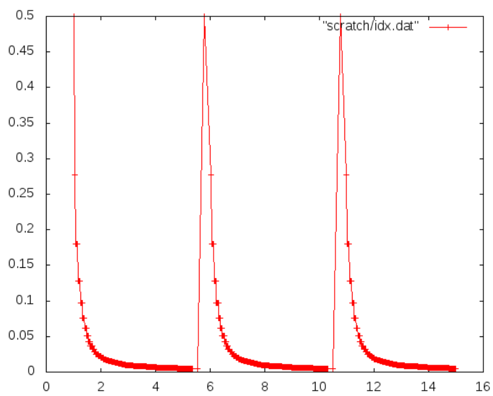


Figure 4.15: The index of the user in a one user routing for 0.005 threshold

because this value of threshold let the ruoter sent more packets.

For a threshold parameter of 0.008, we can see in Figure 4.10 that the maximum number of packets sent by the user is 26.

In Figure 4.11 it is showed the number of packets in the queue versus time. As we said in the no index model, in the first seconds of the simulations and after a packet is dropped, the number of packets in the queue router is not greater than one because the router is able to transmit what the users sends.

The indices of the user of the one user routing implementation can be seen in Figure 4.12. The indices are decreasing, as we described in Chapter 3, so when a packet with an index less or equal to 0.008 that packet is dropped and the number of packets sent goes to zero and the queue gets empty too.

In this case, since we are able to send more packets than in the previous instances, the number of dropping packets is less.

For a threshold value of 0.005, it can be observed in Figure 4.14 that the maximum number of packets in the queue is 25 and this means that when a packet is dropped the reason is the index condition because the maximum size of the queue is never reached.

In Figure 4.13 we can see that the number of packets sent by the user arrives to 36 with a threshold parameter of the router equal to 0.005.

The Figure 4.15 shows that when a packet arrives with an index less or equal to 0.005 the packet is dropped and this causes that the number of packet sent goes to zero and the number of packets in the queue goes to zero also.

The threshold 0.005 case is the instance that there are more packets in the queue and more number of packets sent by the user. Therefore, that make this parameter the better that avoids the congestion of the router.

If instead of a router connected to a server we will have a more difficult implementation, reaching the maximum number of packets in the router will become a problem very difficult to be solved.

In this section we have just seen that applying indices policies is a good way of avoiding future congestion of the window because the number of packets in the router never reaches the maximum. This let us conclude that if we find a threshold parameter in the router to send great amount of packets without having the queue full, we will solve the problem of avoiding the congestion of the network.

Two users implementation

We are going to describe the behavior of two users connected to a router that want to send data to a server. In this implementation, there are more difficulties related to the indices and number of packets send that we are explaining in the following lines.

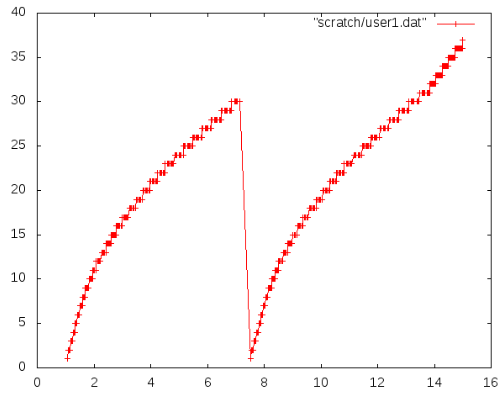


Figure 4.16: The number of packets sent by user0 for no index model and two users

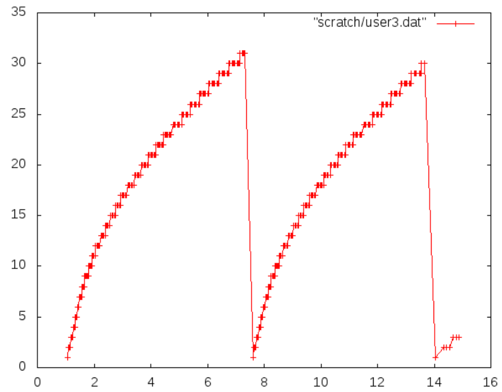


Figure 4.17: The number of packets sent by user1 for no index model and two users

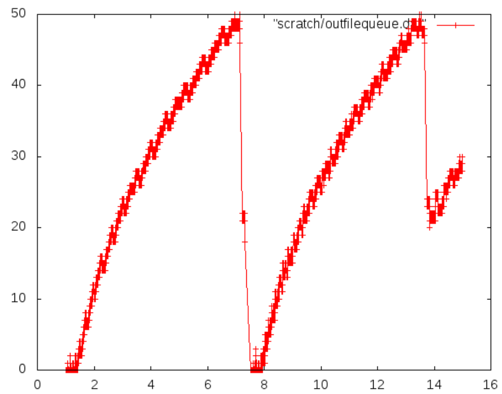
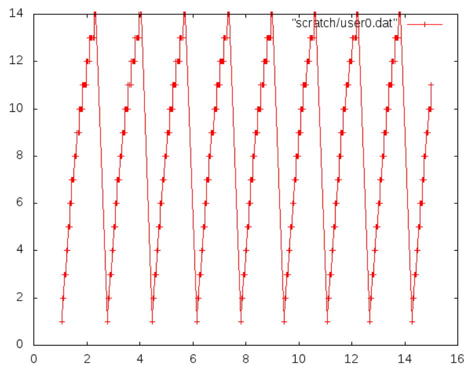
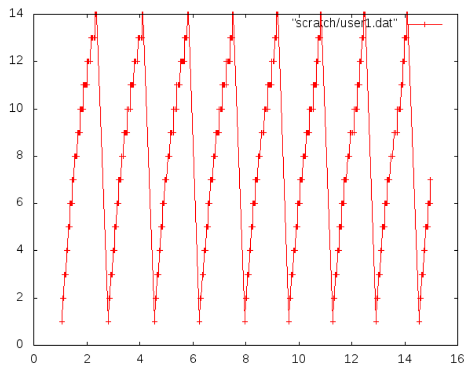


Figure 4.18: The number of packets in the queue of the router in a two users routing for no index model

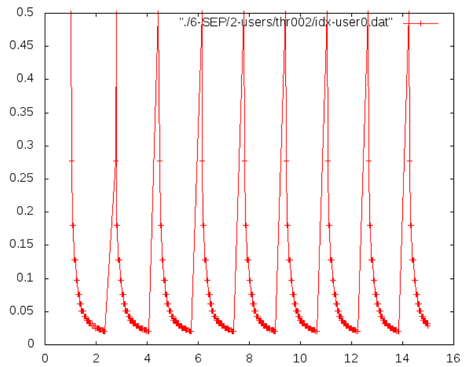


(a) user0

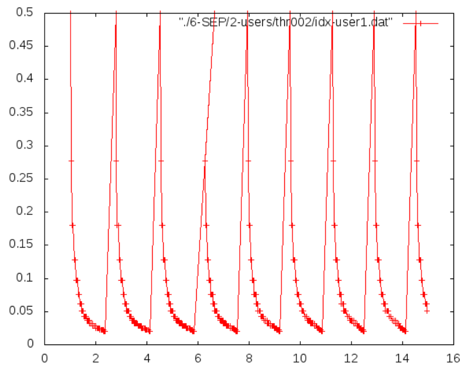


(b) user1

Figure 4.19: Number of packets sent by user0 and user1 in two users routing for0.02 threshold



(a) user0



(b) user1

Figure 4.20: The index of user0 in a two user routing for 0.02 threshold

We will compare no index model with the index model for the following values of threshold: 0.02, 0.01 and 0.008.

We are going to observe the behavior of the two users implementation in no index model showing the following:

- The number of packets sent for each user
- The number of packets in the queue

On the other hand, we will be able to analyze the behavior of the index model for two users implementation showing the following figures:

- The number of packets sent for each user

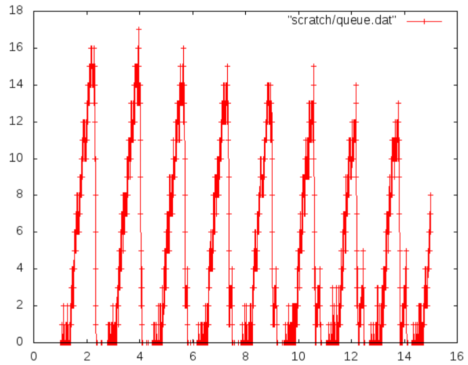
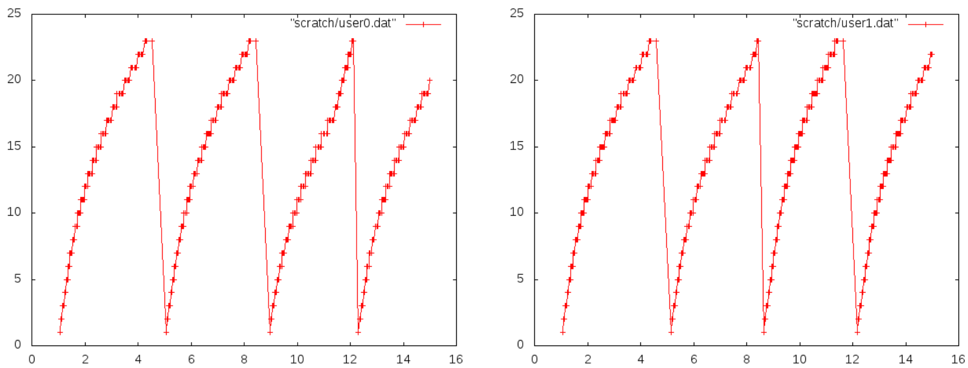


Figure 4.21: The number of packets in the queue of the router in a two users routing for 0.02 threshold



(a) user0 (b) user1

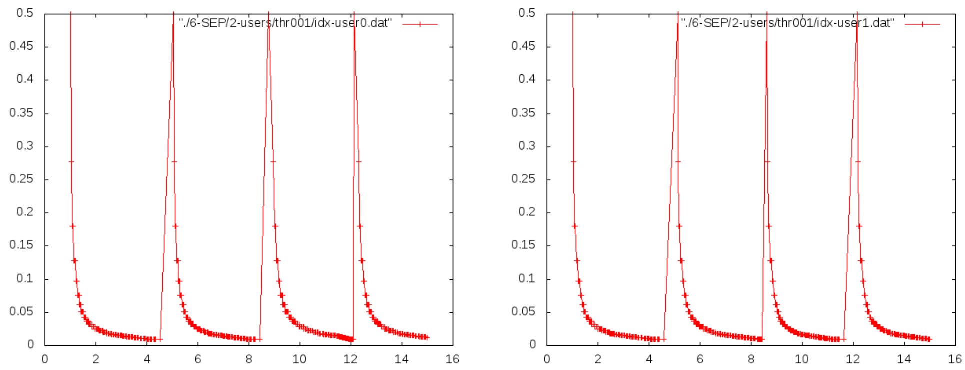
Figure 4.22: Number of packets sent by user0 and user1 in two users routing for 0.01 threshold

- The number of packets in the queue
- The index of the packets that each user is sending

The number of packets in the queue for no index model can be observed in Figure 4.18. In this figure, we can observe that in 7 seconds there is a packet dropped because the maximum number of packets in the queue have been reached.

In Figure 4.16 and in Figure 4.17 we can observe that the number of packets goes to zero in both cases in 7 seconds. That implies that the router has dropped packet of both users in that moment.

When time is equal to 13, however, there is a packet dropped because the number of packets in the queue decreases to the half. But we can observe that only one of the user 's number of packets decreases. That implies that a packet of one user is dropped and



(a) user0 (b) user1

Figure 4.23: The index of user0 in a two user routing for 0.02 threshold

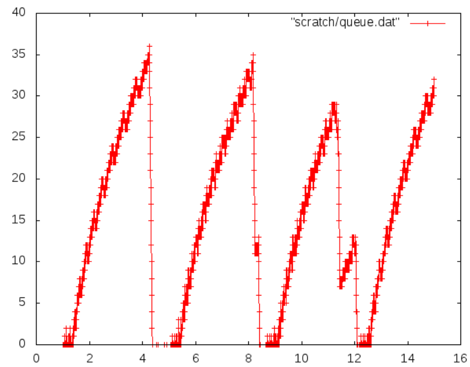


Figure 4.24: The number of packets in the queue of the router in a two users routing for 0.01 threshold

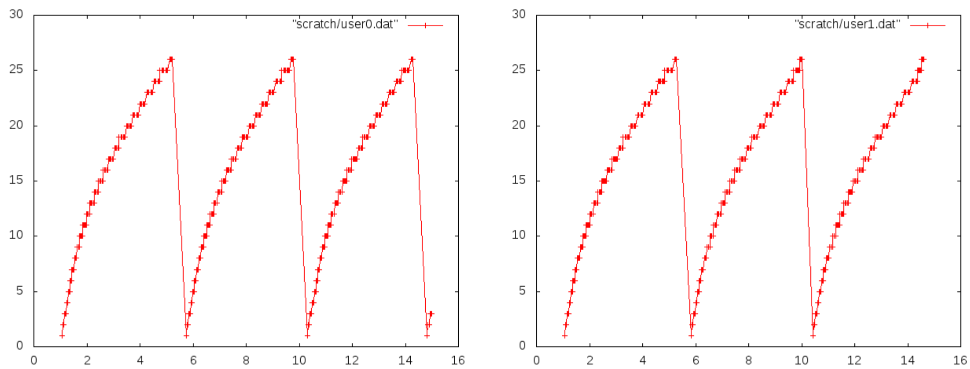
of the other user continues transmitting. So, a user starts sending packets from scratch and the other is able to send faster because the router is capable.

Now, we will start explaining the behavior of this implementation for different values of threshold parameter. After that, we are going to describe the similarities and differences between them and with the no index model.

For a threshold value of 0.002, we observe in 4.19a that the maximum number of packets sent by user0 is 14 and in 4.19b that the maximum number of packets sent by user1 is 14.

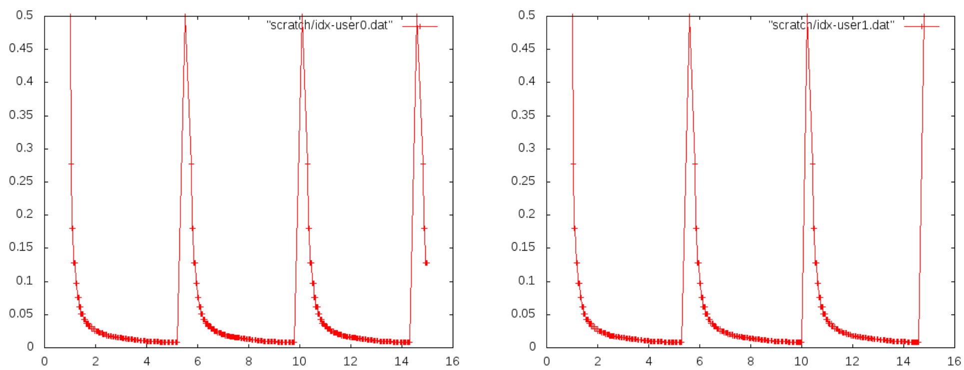
But in this case, the queue behaves differently comparing to the one user implementation, as we can see in Figure 4.21.

The decreasing of the queue size is so high because both users drop packet near in the same time because they have similar characteristics. The maximum value of the queue size is 17 in this instance and this is greater than the one user implementation.



(a) user0 (b) user1

Figure 4.25: Number of packets sent by user0 and user1 in two users routing for 0.008 threshold



(a) user0 (b) user1

Figure 4.26: The index of user0 in a two user routing for 0.02 threshold

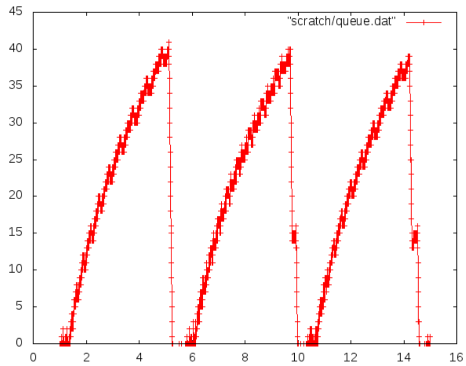


Figure 4.27: The number of packets in the queue of the router in a two users routing for 0.008 threshold

In 4.20a it is showed when the index of the packet arriving to a router that packet is dropped. And in 4.20b happens the same thing.

So, when the router receives a packet of any user with an index less than the threshold parameter 0.02 this packet is dropped. This avoids the congestion of the router because, as we can see the maximum number of packets is never reached.

If we consider a TCP model described above with a threshold parameter of the router equal to 0.01, the maximum number of packets sent by both users is 23 (see 4.22a and 4.22b)

In 4.23a we can see when the router drops a packet of user0 and in 4.23b when in user1. All the drops are due to the index condition and never happens that the queue is full, as it it showed in Figure 4.24.

For a threshold parameter of 0.008, we can see in 4.25a and in 4.25b that the maximum number of packets sent by user0 and user1 is 26.

In 4.26a we can observe the indices of the packet that user0 is sending in time and in 4.26b the indices of user1. In this figures we can know when a packet is dropped because of the index condition.

In Figure 4.27 it is showed the number of packets in the queue versus time. Here we can observe that the router is never congested in this case, but the number of packets in the queue is near to the maximum. Besides, this parameter let the users send more packets than the others, so we can conclude that the value of 0.008 is the best threshold parameter for two users implementation.

4.4 Conclusions

The Network Simulator-3 package have helped us to check that the indices policies can be applied in internet routing.

The results obtained in one user and two users implementation let us conclude that there is a limited congestion of the flow/packet in the network if we choose correct values of the threshold.

The simulations we have run let us know that what are the threshold parameter that allow the users sends more packets.

The main part of the analyzing has been the queue because that the decision is taken there. But we are also interested on how the indices change in the received packets because that is the reason of transmitting or not transmitting the flow.

The threshold is also a very important parameter that we have to choose carefully, because if it is not high enough the velocity of sending packets is small. But if we choose a big threshold parameter, congestion of the flow/packet can occur.

Now, we have a code that simulates additive increasing and multiplicative decreasing TCP. We will investigate more complex scenarios in future work.

Chapter 5

Conclusions and Future Work

In this thesis we have developed a model for the important problem of congestion avoidance in data networks.

We formulate the additive increasing and multiplicative decreasing TCP model as a Markov Decision Process. In this work we use indices policies in a router to take a decision of transmitting or not transmitting the flow, taking into account α-fairness criterion.

The index policies are proven that there are simple to implement and provide a good solution of the maximum problem presented above. In this work, we verify that indices policies are applicable in TCP routing work for additive increase and every decrease model.

From the theoretical point of view, we show that the model under consideration is always indexable and can be solved under threshold policies in no restarting case under certain conditions of α parameter.

We conclude from numerical results that TCP is always indexable for any decrease factor. However, it can be solved under threshold policies only for some values of α parameter, depending on the decrease factor we are using. This is good news for possible implementation of index policies.

We have carried out extensive ns-3 simulations that show that applying index policies is an simple applicable way that limits the congestion of the network, because the queue is never full in the examples we have checked.

To finish, we have a code that simulates additive increasing and multiplicative decreasing instances of TCP.

5.1 Future Work

There are various different directions our work can be taken further. From the theoretical point of view, we have:

- Slow-start and other TCP models can be considered in our modeling and can be tested the behavior of index policies in internet router for this implementation.
- We assume the users that want to send data to a server are honest and they send the correct value of index. However, the router can estimate the value of the index with the information that reaches with the packets, according to the values of index obtained in the theoretical section of this work
- Additional research can be carried out involving this modeling of TCP for internet routers. More complicated topologies can be formed to implement this model.

From the practical point of view we have identified the following problems:

- Multiplicative decrease instances are the most common used in TCP applications. In this work we only have tested the behavior of TCP Tahoe implementation, but we are interested on how the model behaves in this cases.
- We believe that the efficiency of index policies for a greater number of users is higher and ns-3 simulator is the tool we are going to use to verify it.
- The router can take the decision of dropping packet in a different way than the threshold condition. One possible way of implementing that is dropping the packet of the user with smallest index when the router is near to be congested.
- We have simulated all the examples for a drop-tail implementation. We would like to model and test different buffer management policies.

References

Altman, E., Avrachenkov, K., and Garnaev, A. (2008). Generalized α-fair resource allocation in wireless networks. *IEEE Conference: Decission and Control*.

Avrachenkov, K., Ayesta, U., Brown, P., and E., N. (2001). The effect of router buffer size on the tcp performance. *LONIIS*.

Avrachenkov, K., Ayesta, U., Brown, P., and Nyberg, E. (2004). Differentiation between short and long tcp flows: predictability of the response time. *SIGCOMM*.

Jacko, P. and Sanso, B. (2010). Optimal anticipative congestion control of flows with time-varying input stream. *Performance Evaluation*.

Jacobson, V. (1988). Congestion avoidance and control. *SIGCOMM*.

Niño Mora, J. (2002). Dynamic allocation indices for restless projects and queueing admission control: a polyhedral approach. *Mathematics on Operation Research*.

Puterman, M. L. (1994). *Markov decision processes: discrete stochastic dynamic programming*. Springer.

Riley, G. and Henderson, T. (2010). *The ns-3 Network Simulator*. Springer.